

Infoprint Server for iSeries



# User's Guide

*Version 5, Release 2.0*



Infoprint Server for iSeries



# User's Guide

*Version 5, Release 2.0*

**Note**

Before using this information and the product it supports, be sure to read the general information in "Notices" on page 165.

**Third Edition (January 2003)**

This edition applies to the IBM Infoprint Server for iSeries Version 5 Release 2 Modification 0 licensed program, Program Number 5722-IP1, and to all subsequent releases and modifications until otherwise indicated in new editions or technical newsletters. Be sure to use the correct edition for the level of the product. This edition replaces G544-5775-01.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

The IBM Printing Systems Division welcomes your comments. A form for reader's comments is provided at the back of this publication. If the form has been removed, you may send your comments to the following address:

INFORMATION DEVELOPMENT  
THE IBM PRINTING SYSTEMS DIVISION  
DEPARTMENT H7FE BUILDING 004M  
PO BOX 1900  
BOULDER CO 80301-9191  
U.S.A.

If you prefer to send comments electronically, use one of the following methods:

- Internet: [printpub@us.ibm.com](mailto:printpub@us.ibm.com)
- Fax: 1-800-524-1519 or 1-303-924-6873

**Internet**

Visit our home pages at <http://www.ibm.com/printers>  
and <http://www.ibm.com/servers/eserver/iseries/>.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 2001, 2003. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

---

# Contents

<b>Figures</b>	vii
<b>About Infoprint Server for iSeries: User's Guide (G544–5775)</b>	ix
Who Should Read this Book	ix
Conventions and Terminology Used in This Book	x
Understanding Syntax Notation	x
Prerequisite and Related information	xi
iSeries Navigator	xii
<b>Summary of Changes</b>	xiii
<b>Chapter 1. Introducing Infoprint Server</b>	1
How Infoprint Server Fits in with OS/400 Printing	1
What's New in Infoprint Server Version 5.2?	5
Installing Infoprint Server	6
<b>Chapter 2. Using the Create AFP Data (CRTAFPDTA) Command</b>	7
Overview of CRTAFPDTA	7
Syntax	9
Planning for CRTAFPDTA	11
Authorities	11
Indexing with CRTAFPDTA	12
Indexing with Data Values	13
Indexing with Literal Values	13
Indexing Limitations	13
Retrieving Resources	14
How CRTAFPDTA Searches for AFP Resources	14
CRTAFPDTA Parameters	15
CRTAFPDTA Examples	25
Merging Resources with Data	26
Indexing with CRTAFPDTA	26
E-mailing the CRTAFPDTA Output	32
How it Works	32
Steps to E-mail the Output	32
Example	33
E-mail Considerations	34
CRTAFPDTA and ACIF	34
<b>Chapter 3. Using the GIF to AFP Transform</b>	37
Syntax	37
Overview of gif2afp	37
Output Data Types	38
Scaling	38
Halftoning	39
Installing the Image Transforms	39
Planning for gif2afp	39
Setting Up a Configuration File	39
Using Environment Variables	40
Editing Your Windows Path	41
Authorities	41
Parameters	41
Transforming GIF Data to AFP	51
Input to the GIF to AFP Transform	51

Specifying the Output File . . . . .	51
Invoking gif2afp . . . . .	52
Command Line Arguments . . . . .	52
Examples . . . . .	52
Error Recovery . . . . .	53
Limitations . . . . .	54
Acknowledgments . . . . .	54
 <b>Chapter 4. Using the JPEG to AFP Transform . . . . .</b>	<b>55</b>
Syntax . . . . .	55
Overview of jpeg2afp . . . . .	55
Output Data Types . . . . .	56
Scaling . . . . .	56
Halftoning . . . . .	57
Installing the Image Transforms . . . . .	57
Planning for jpeg2afp . . . . .	57
Setting up a Configuration File . . . . .	57
Using Environment Variables . . . . .	58
Editing Your Windows Path . . . . .	59
Authorities . . . . .	59
Parameters . . . . .	59
Transforming JPEG Data to AFP . . . . .	68
Input to the JPEG to AFP Transform . . . . .	68
Specifying the Output File . . . . .	68
Invoking jpg2afp . . . . .	69
Command Line Arguments . . . . .	69
Examples . . . . .	69
Error Recovery . . . . .	70
Limitations . . . . .	71
Acknowledgments . . . . .	71
 <b>Chapter 5. Using the TIFF to AFP Transform . . . . .</b>	<b>73</b>
Syntax . . . . .	73
Overview of tiff2afp . . . . .	73
Output Data Types . . . . .	74
Scaling . . . . .	74
Halftoning . . . . .	75
Installing the Image Transforms . . . . .	75
Planning for tiff2afp . . . . .	75
Setting up a Configuration File . . . . .	75
Using Environment Variables . . . . .	76
Editing Your Windows Path . . . . .	77
Authorities . . . . .	77
Parameters . . . . .	77
Transforming TIFF Data to AFP . . . . .	88
Input to the TIFF to AFP Transform . . . . .	88
Specifying the Output File . . . . .	88
Invoking tiff2afp . . . . .	89
Command Line Arguments . . . . .	89
Examples . . . . .	89
Error Recovery . . . . .	90
Limitations . . . . .	90
Acknowledgments . . . . .	91
 <b>Chapter 6. Using the PDF Subsystem. . . . .</b>	<b>93</b>
Overview . . . . .	93

Default Transform Characteristics . . . . .	94
Before Using the PDF Subsystem . . . . .	95
Setting up Your PSF Configuration Object . . . . .	95
Configuring the Device . . . . .	96
Varying the Device ON . . . . .	96
Starting the Writer. . . . .	97
Transforming Data . . . . .	97
Example . . . . .	97
Accessing the PDF Subsystem with iSeries Access . . . . .	98
Modifying the Input Data . . . . .	99
Using DDS to Modify the Input File . . . . .	99
Using AFP Toolbox to Modify the Input File . . . . .	100
Using CRTAFPDTA to Modify the Input File . . . . .	100
PSF Configuration Object Parameters . . . . .	100
Usage Notes . . . . .	106
Bar Codes . . . . .	107
Color . . . . .	107
Error Checking . . . . .	107
Find Function . . . . .	107
Fonts . . . . .	107
Other Unsupported IPDS Functions . . . . .	108
Output File Size . . . . .	108
Page Numbering . . . . .	109
Printing Considerations . . . . .	109
Use of Printer Features . . . . .	109
Viewing Considerations . . . . .	110
Error Recovery . . . . .	110
Messages . . . . .	110
Troubleshooting . . . . .	111
PDF Conversion Completion Data Queue. . . . .	112
Using the Data Queue. . . . .	112
Format . . . . .	113
 <b>Chapter 7. Transforming PCL, PDF, and PostScript Data to AFP . . . . .</b>	 115
Overview. . . . .	115
Syntax . . . . .	116
Planning for the PCL, PDF, or PostScript to AFP Transforms . . . . .	116
Authorities . . . . .	117
System Requirements . . . . .	117
Creating the Work Directory. . . . .	117
Configuring Transform Manager . . . . .	117
Customizing the PostScript to AFP and PDF to AFP Transforms . . . . .	120
Starting and Ending Transform Manager . . . . .	121
Setting Up the Printer Device Description. . . . .	122
Transforming Data . . . . .	122
Deactivating and Reactivating the Transforms . . . . .	123
Steps to Deactivate the Transforms . . . . .	123
Reactivating the Transforms . . . . .	124
Font Downloader. . . . .	124
Syntax . . . . .	125
Troubleshooting . . . . .	126
 <b>Chapter 8. Sending E-mail. . . . .</b>	 129
Steps to Send a PDF File as E-mail . . . . .	130
Enabling Your OS/400 to Send E-Mail . . . . .	130
Steps to Enable Your OS/400 to Send E-mail . . . . .	130

Setting up the OS/400 as a POP E-mail Client . . . . .	132
Creating a PSF Configuration Object . . . . .	133
Using Mail Tags . . . . .	133
Inserting a Mail Tag in the Printer File . . . . .	133
Inserting a Mail Tag in the Spooled File . . . . .	134
Inserting a Mail Tag in the Input File . . . . .	134
Using AFP Manager to Send E-mail. . . . .	135
Before You Begin . . . . .	135
Steps to Create a PDF File and Send it as E-mail . . . . .	135
Checking for Non-delivery Messages . . . . .	136
Setting up a POP3 Client for a SNDDST Sender . . . . .	136
Setting QUTCOFFSET . . . . .	137
 <b>Chapter 9. Related Products</b> . . . . .	 139
iSeries Access . . . . .	139
Overview . . . . .	139
iSeries Access for Web . . . . .	139
Infoprint Designer for iSeries . . . . .	140
 <b>Appendix A. Related Tasks</b> . . . . .	 141
Turning a PC AFP Resource into an OS/400 Resource. . . . .	141
 <b>Appendix B. PDF Mapping Program</b> . . . . .	 143
Parameters . . . . .	144
E-mail Input information . . . . .	145
E-mail Output . . . . .	147
Extension Area Format . . . . .	149
Stream File Format . . . . .	152
Individual Stream File Information Format . . . . .	153
 <b>Appendix C. Templates</b> . . . . .	 155
PDF Mapping Program in C. . . . .	155
PDF Mapping Program in RPG . . . . .	156
Exit Program in C . . . . .	159
 <b>Appendix D. Exit Points</b> . . . . .	 163
 <b>Notices</b> . . . . .	 165
Trademarks. . . . .	166
 <b>Glossary</b> . . . . .	 169
 <b>Bibliography</b> . . . . .	 177
Infoprint Server . . . . .	177
Advanced Function Presentation (AFP) . . . . .	177
iSeries Access . . . . .	177
Infoprint Designer . . . . .	177
OS/400 . . . . .	178
Printers . . . . .	178
PrintSuite . . . . .	178
Redbooks . . . . .	178
TCP/IP . . . . .	178
 <b>Index</b> . . . . .	 181



---

## Figures

1. OS/400 syntax example . . . . .	xi
2. iSeries Printing without Infoprint Server . . . . .	2
3. iSeries Output Management with Infoprint Server . . . . .	4
4. Infoprint Server Components . . . . .	5
5. Using CRTAFPDTA . . . . .	8
6. CRTAFPDTA command syntax . . . . .	10
7. AFP Document with Index Tags and the Index Object File. . . . .	12
8. Example CRTAFPDTA Command. . . . .	26
9. Example of a Customer's Phone Bill. . . . .	28
10. Example Telephone Bill Input File. . . . .	29
11. Example CRTAFPDTA Command. . . . .	30
12. gif2afp command syntax . . . . .	37
13. jpeg2afp command syntax . . . . .	55
14. tiff2afp command syntax . . . . .	73
15. Transforming a Spooled File to PDF. . . . .	94
16. Transforming PCL, PDF, and PostScript data to AFP Data. . . . .	116
17. Starting Transform Manager . . . . .	116
18. Ending Transform Manager . . . . .	116
19. Default Configuration File . . . . .	119
20. Font Downloader Process . . . . .	125
21. Starting Font Downloader . . . . .	125
22. Ending Font Downloader . . . . .	125



---

## About Infoprint Server for iSeries: User's Guide (G544–5775)

This publication provides information about using Infoprint® Server for iSeries™ Version 5 Release 2 (licensed program number 5722–IP1), hereafter referred to as Infoprint Server. The term iSeries refers to the server previously called AS/400®. The term OS/400® refers to the operating system of the iSeries.

You can use this publication both as a guide and as a reference to help you learn about the following:

- Chapter 1, “Introducing Infoprint Server” on page 1 gives an overview of this product and describes how it fits in with OS/400 printing.
- Chapter 2, “Using the Create AFP Data (CRTAFPDTA) Command” on page 7 describes how to use the CRTAFPDTA command.
- Chapter 3, “Using the GIF to AFP Transform” on page 37 describes how to use the gif2afp command.
- Chapter 4, “Using the JPEG to AFP Transform” on page 55 describes how to use the jpeg2afp command.
- Chapter 5, “Using the TIFF to AFP Transform” on page 73 describes how to use the tiff2afp command.
- Chapter 6, “Using the PDF Subsystem” on page 93 describes how to transform a spooled file to the Portable Document Format (PDF) data stream
- Chapter 7, “Transforming PCL, PDF, and PostScript Data to AFP” on page 115 describes how to transform Printer Control Language (PCL), PDF, and PostScript data to AFP™ format.
- Chapter 8, “Sending E-mail” on page 129 describes how to use Infoprint Server to e-mail PDF files.
- Chapter 9, “Related Products” on page 139 describes how to use the AFP Viewer and iSeries Access in conjunction with Infoprint Server. It also describes iSeries Access for Web.
- Appendix A, “Related Tasks” on page 141 describes how to use OS/400 commands to turn an AFP resource on your PC into an OS/400 resource.
- Appendix B, “PDF Mapping Program” on page 143 describes how to use the e-mail exit program and the PDF transform exit program in conjunction with Infoprint Server.
- Appendix C, “Templates” on page 155 contains templates for e-mail exit programs.
- Appendix D, “Exit Points” on page 163 lists the exit points that Infoprint Server registers.
- The Glossary contains terms and definitions related to PSF and the printing environment.
- The bibliography contains selected titles and order numbers of IBM® publications related to PSF and the printing environment.

---

## Who Should Read this Book

The information in this publication is directed at people in applications and operational functions that need to understand various functional components of Infoprint Server and how to implement them in their environment.

This publication is written with the assumption that you have experience with application programming and with Advanced Function Presentation™ (AFP) printers. It is also assumed that you are familiar with the system and using it to print.

---

## Conventions and Terminology Used in This Book

In this manual, the term AFP Viewer refers to both the AFP Workbench Viewer, available with iSeries Access, and the AFP Viewer Plug-in, available as a free download from IBM.

## Understanding Syntax Notation

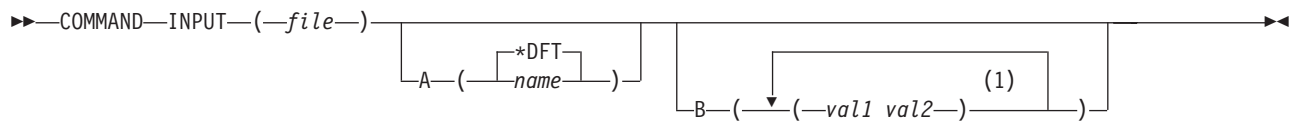
The following rules apply to coding illustrations and Windows® command syntax throughout this publication:

- Variable data is printed in italics. Enter specific data to replace the characters in italics. For example, for *ImageType* you could enter IM1. Italics also identify the names of publications.
- Bold highlighting identifies commands, keywords, files, directories, and other items whose names are predefined by the system, or items that must be entered as is, such as **-ink**.
- Monospacing, for example, gif2afp identifies examples of specific data values, examples of text similar to what you might see displayed, examples of portions of program code similar to what you might write as a programmer, messages from the system, or information you should actually type.
- Do not enter the following symbols as part of a parameter or option:
  - Vertical Bar |
  - Underscore \_\_\_\_
  - Brackets [ ]
  - Braces { }
  - Ellipsis ...
- A vertical bar between two values means that you select only one of the values. For example, VALUE1 | VALUE2 means that you can choose either VALUE1 or VALUE2.
- An underscored value means that if an option is not specified, the underscored value, called the default, is used. For example, Print (yes | no) means that if you do not specify anything for Print, its value is set to yes.
- Brackets around a value mean that you do not have to select the value; the value is optional. For example, File [library] means you can specify the library but you do not have to.
- Braces around a value mean that you must select one of the mutually exclusive values. For example, { THIS | THAT } means you must specify either THIS or THAT.
- An ellipsis following a command or set of commands indicates the command or set of commands can be repeated. For example, Print (*file...*) means that you can specify multiple values for *file*.

The following rules apply to OS/400 command syntax throughout this publication:

- Variable data is printed in italics. Enter specific data to replace the characters in italics. For example, for *indexing-page-limit* you could enter 50. Italics also identify the names of publications.
- Commands are printed in all upper case letters. They must be entered exactly as they appear.

- Values are enclosed in parenthesis, for example, (*file*). Enter the parenthesis as part of the parameter.
- A command or keyword printed on the baseline is required. For example, in Figure 1, COMMAND and INPUT (*file*) are required, but A and B are optional.
- A value printed above the baseline is the default. For example, in Figure 1, \*DFT is the default value for the keyword A, but B does not have a default value.
- If values are printed above one another, you can choose only one of the values. For example, in Figure 1, you can choose \*DFT or *name* as the value for A.
- If values are printed next to each other, you can use all of the values. For example in Figure 1, you can enter either or both *val1* and *val2* as values of B.
- If values, commands, or keywords are printed with an arrow above them, you can repeat the part under the arrow. For example, in Figure 1, you can enter up to four values of *val1* and *val2* for B.



#### Notes:

- 1 A maximum of four repetitions.

Figure 1. OS/400 syntax example

## Prerequisite and Related information

Use the iSeries Information Center as your starting point for looking up iSeries technical information.

You can access the Information Center two ways:

- From the following Web site:  
<http://www.ibm.com/servers/eserver/iseries/infocenter>
- From CD-ROMs that ship with your order:  
*iSeries Information Center*, SK3T-4091-02. This package also includes the PDF versions of iSeries manuals, *iSeries Information Center: Supplemental Manuals*, SK3T-4092-01, which replaces the Softcopy Library CD-ROM.

The iSeries Information Center contains advisors and important topics such as Java™, TCP/IP, Web serving, secured networks, logical partitions, clustering, CL commands, and system application programming interfaces (APIs). It also includes links to related IBM Redbooks™ and Internet links to other IBM Web sites such as the IBM home page.

With every new hardware order, you receive the *iSeries Setup and Operations CD-ROM*, SK3T-4098-01. This CD-ROM contains IBM @server iSeries Access for Windows and the EZ-Setup wizard. iSeries Access offers a powerful set of client and server capabilities for connecting PCs to iSeries servers. The EZ-Setup wizard automates many of the iSeries setup tasks.

The Printing Systems iSeries Products Web page contains information about this product. See this Web page:

<http://www.ibm.com/printers/R5PSC.NSF/Web/as400overview>

For other related information, see the “Bibliography” on page 177.

## **iSeries Navigator**

IBM iSeries Navigator is a powerful graphical interface for managing your iSeries servers. iSeries Navigator functionality includes system navigation, configuration, planning capabilities, and online help to guide you through your tasks. iSeries Navigator makes operation and administration of the server easier and more productive and is the only user interface to the new, advanced features of the OS/400 operating system. It also includes Management Central for managing multiple servers from a central system.

You can find more information on iSeries Navigator in the iSeries Information Center and at the following Web site:

<http://www.ibm.com/eserver/iseries/navigator/>

---

# Summary of Changes

## Summary of Changes for Infoprint Server for iSeries: User's Guide, G544-5775-02.

This publication contains additions and changes to information that was previously presented in *Infoprint Server for iSeries: User's Guide*, G544-5775-01, which supports Infoprint Server for iSeries Version 5 Release 2.0. The technical additions and changes are marked with a change bar ( | ) in the left margin.

The following information is new or updated:

- You might not be able to print the PDF generated by the PDF subsystem on your PDF printer. For printer requirements, see "Printing Considerations" on page 109 in Chapter 6, "Using the PDF Subsystem".
- More solutions to common problems have been added to "Troubleshooting" on page 111 in Chapter 6, "Using the PDF Subsystem".
- "Deactivating and Reactivating the Transforms" on page 123 has been added to Chapter 7, "Transforming PCL, PDF, and PostScript Data to AFP".
- "Increasing the Maximum Memory Setting" on page 121 has been added to Chapter 7, "Transforming PCL, PDF, and PostScript Data to AFP".
- More solutions to common problems have been added to "Troubleshooting" on page 126 in Chapter 7, "Transforming PCL, PDF, and PostScript Data to AFP".
- "Steps to Enable Your OS/400 to Send E-mail" on page 130 has been updated.
- The steps to accomplish tasks related to, but not requiring, Infoprint Server have been removed from Chapter 9, "Related Products" on page 139. These steps can be found in *iSeries Guide to Advanced Function Presentation*, the iSeries Information Center, or the iSeries Navigator online help.





---

## Chapter 1. Introducing Infoprint Server

Infoprint Server for iSeries (hereafter referred to as Infoprint Server) is a separately orderable program for OS/400 Version 5 Release 1 and higher. It provides essential electronic delivery options for iSeries along with the capabilities to consolidate network printing on the iSeries.

---

### How Infoprint Server Fits in with OS/400 Printing

Infoprint Server significantly expands the possibilities of print and electronic output in an iSeries environment, integrating new capabilities within the existing print framework. Let's take a closer look and see how. The existing output subsystem is generally print-centric. Figure 2 on page 2 depicts the flow within this subsystem. Any OS/400 application that creates print uses a printer file to provide job-level control information as the data is written to the output queue. DDS keywords can be used to define how application and static information is placed on each page. With these instructions, the application places print data (a spooled file) on an OS/400 output queue. The print data on the queue is usually either SCS (SNA Character Set) for simple line-mode output or AFP with graphics. With AFP print data, there are frequently embedded references to external print resources, such as fonts, overlays, and page segments.

Once the print data resides on the OS/400 output queue, there are three drivers that can route the print data to a printer. The driver is automatically selected based on the type of target printer. Base OS/400 print management includes the driver for printing to SCS printers. Host Print Transform is the driver for ASCII printers, principally PCL printers. When a PCL printer is selected by the print writer, the print file (either SCS or AFP) is passed to Host Print Transform. Host Print Transform then transforms the print data into ASCII and sends it to the printer. When an Intelligent Printer Data Stream™ (IPDS™) printer is started, Print Services Facility™ (PSF) for OS/400 is automatically invoked and interactively manages the entire printing process with the printer. When PSF for OS/400 manages AFP print data, it ensures that any external resources required, such as fonts, overlays, and images, are in printer memory when needed.

With OS/400 3.2 and 3.7, two additional types of external resources appeared on the scene: page definitions and form definitions. These resources are a standard part of the AFP architecture and enable the pages to be formatted independently of the application program. Infoprint Designer for iSeries is a fully graphical output composition system that uses these resources to design new applications or re-engineer existing ones.

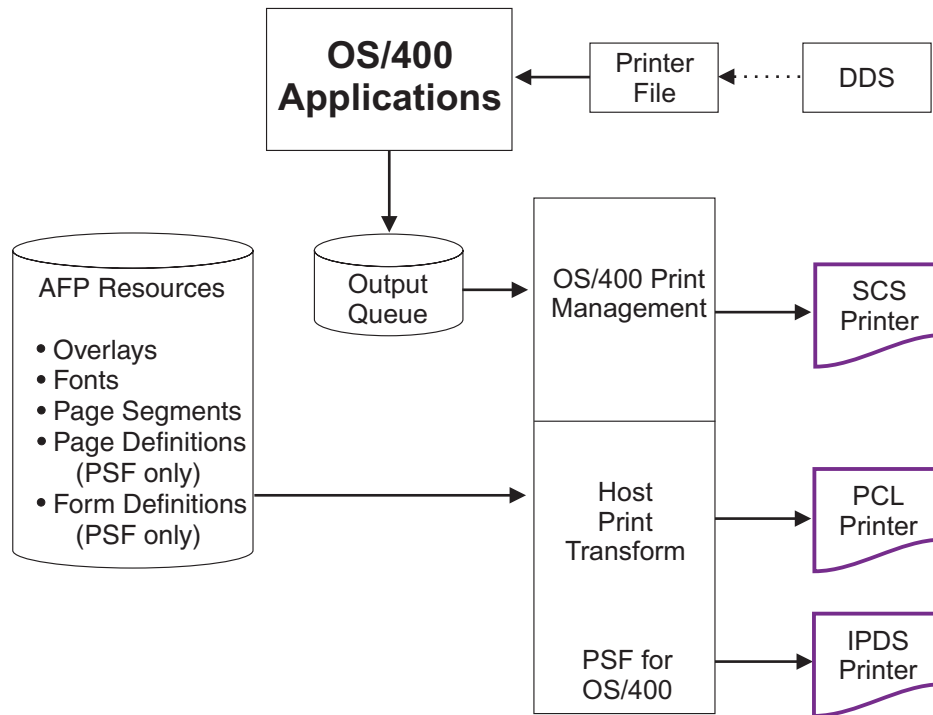


Figure 2. iSeries Printing without Infoprint Server

Infoprint Server opens up this print architecture. Figure 3 on page 4 shows the functional elements that Infoprint Server adds to the picture. The Infoprint Server components are shaded. The general focus is in two major areas: (1) projecting OS/400 output to the network, and (2) incorporating the network into the OS/400 print functions.

Infoprint Server has five functional components:

- PDF services
- E-mail iSeries output
- PDF, PostScript, and PCL to AFP datastream transforms
- Create AFP Data command for AFP indexing and creation of portable AFP
- Image transforms for GIF, TIFF, and JPEG to iSeries format

Let's go through the Infoprint Server functional elements one by one. First, transforms have been built into OS/400 that convert several types of input data to image-based AFP and place it on an OS/400 output queue. The input data can be Printer Control Language (PCL), PostScript, or Portable Document Format (PDF) print data. This enables most ASCII output created in the OS/400 or on the network to be put in native OS/400 format (AFP), which lets users take advantage of the OS/400 print management capabilities. These transforms are managed by Transform Manager. For more information about the PDF, PCL, and PostScript to AFP transforms, see Chapter 7, "Transforming PCL, PDF, and PostScript Data to AFP" on page 115.

A central component of Infoprint Server is the PDF subsystem. This subsystem enables the conversion of any standard OS/400 output (SCS, AFP, mixed data, IPDS, or OfficeVision/400™) to Adobe PDF. The conversion process is highly integrated and creates a native, text-based ASCII PDF data file when the input to the transform is non-image. This PDF file can then be routed to one of three places: an integrated file system directory, an output queue, or outbound by e-mail. In

addition, you can use AFP Toolbox for AS/400, the Create AFP Data command, or DDS keywords to logically segment a print file. This logical segmentation carries forward to the PDF transform process. This provides the ability to create a single PDF file with indexing information or multiple PDF files from a single input file. PDF files stored in the integrated file system are available to any client, network, or Web application. For more information about the PDF Subsystem, see Chapter 6, “Using the PDF Subsystem” on page 93.

Automatic e-mail distribution is a key additional function integrated into the PDF subsystem. For any standard print file in an OS/400 output queue, you can specify that the data be converted to PDF and sent as e-mail. A user exit has been added to further customize the e-mail process. For example, you can select an input file with logical segments that are built in. Each logical segment is transformed into a PDF file and the user exit lets you link to an address database for the e-mail addresses for each PDF file. For more information about e-mailing a PDF file, see Chapter 6, “Using the PDF Subsystem” on page 93 and Chapter 8, “Sending E-mail” on page 129.

PDF is one industry-standard approach to electronic distribution of output. AFP is another. There is an AFP Viewer application built into iSeries Access and an AFP Viewer plug-in is available for Netscape Navigator or Internet Explorer. Because AFP files can have external resources, it is important that any portable AFP data is sent with those external resources. For Web use, Infoprint Server provides the ability to convert an AFP input file to fully portable format. The resources are embedded within the data. In addition, Infoprint Server with its Create AFP Data (CRTAFPDTA) command, can add indexing within this portable file. This facilitates easy navigation by the person viewing the data. The Create AFP Data (CRTAFPDTA) command can also convert line (\*LINE) or mixed (\*AFPDSLNE) data to fully-resolved AFP data for printing or viewing. For more information about the Create AFP Data (CRTAFPDTA) command, see Chapter 2, “Using the Create AFP Data (CRTAFPDTA) Command” on page 7.

Another element of Infoprint Server is its image transforms. These transforms, which execute on the client, convert industry standard image formats (GIF, TIFF, or JPEG) into AFP page segments or overlays on your PC. You can use the AFP Manager component of iSeries Navigator or OS/400 commands to create the page segment or overlay on your OS/400. This lets you use an image in a client or network application and then embed it in an OS/400 application. For more information about the image transforms, see Chapter 3, “Using the GIF to AFP Transform” on page 37, Chapter 4, “Using the JPEG to AFP Transform” on page 55, and Chapter 5, “Using the TIFF to AFP Transform” on page 73.

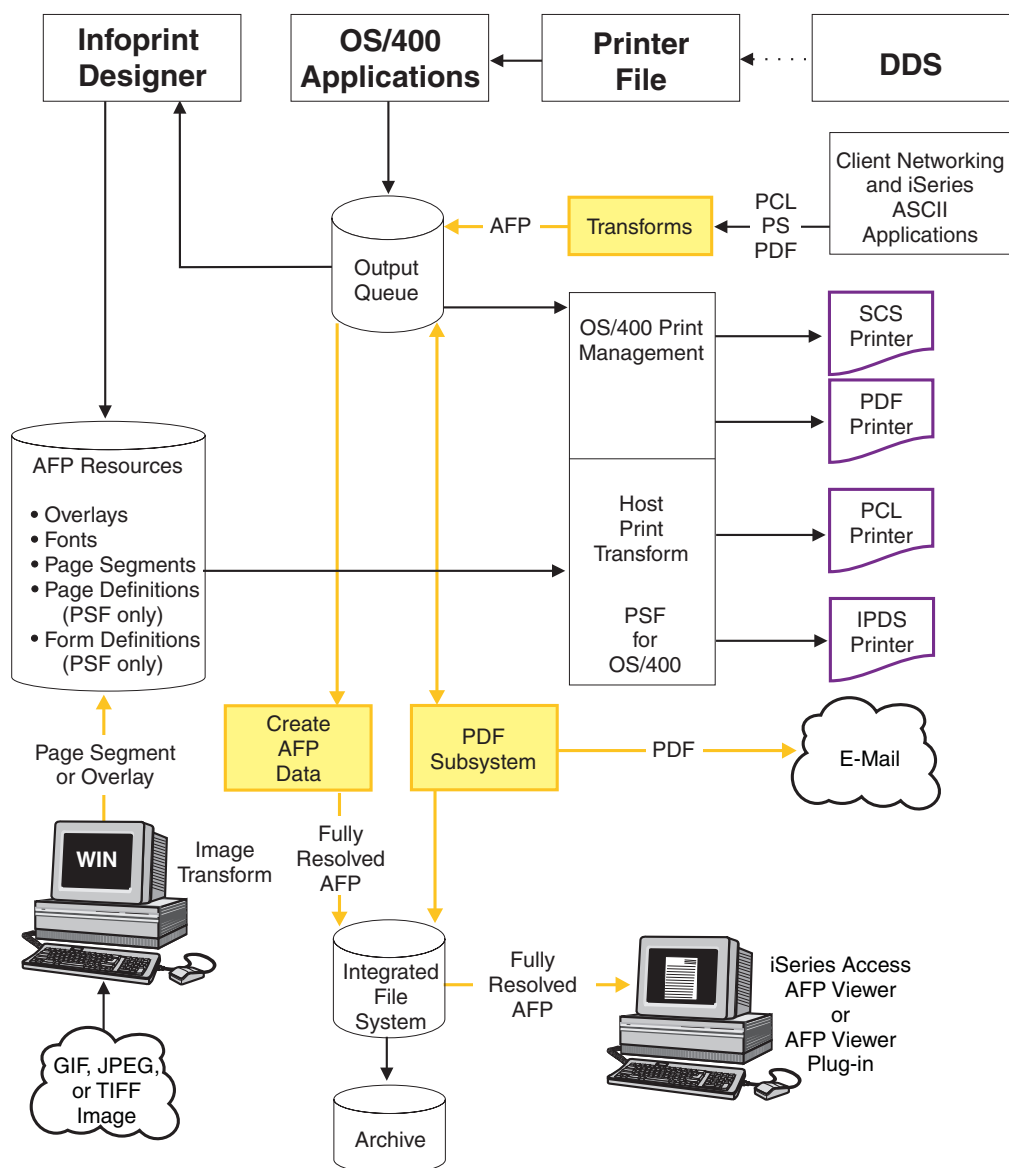


Figure 3. iSeries Output Management with Infoprint Server

Figure 4 on page 5 shows how each Infoprint Server component fits into your OS/400 system. The Infoprint Server components are shaded.

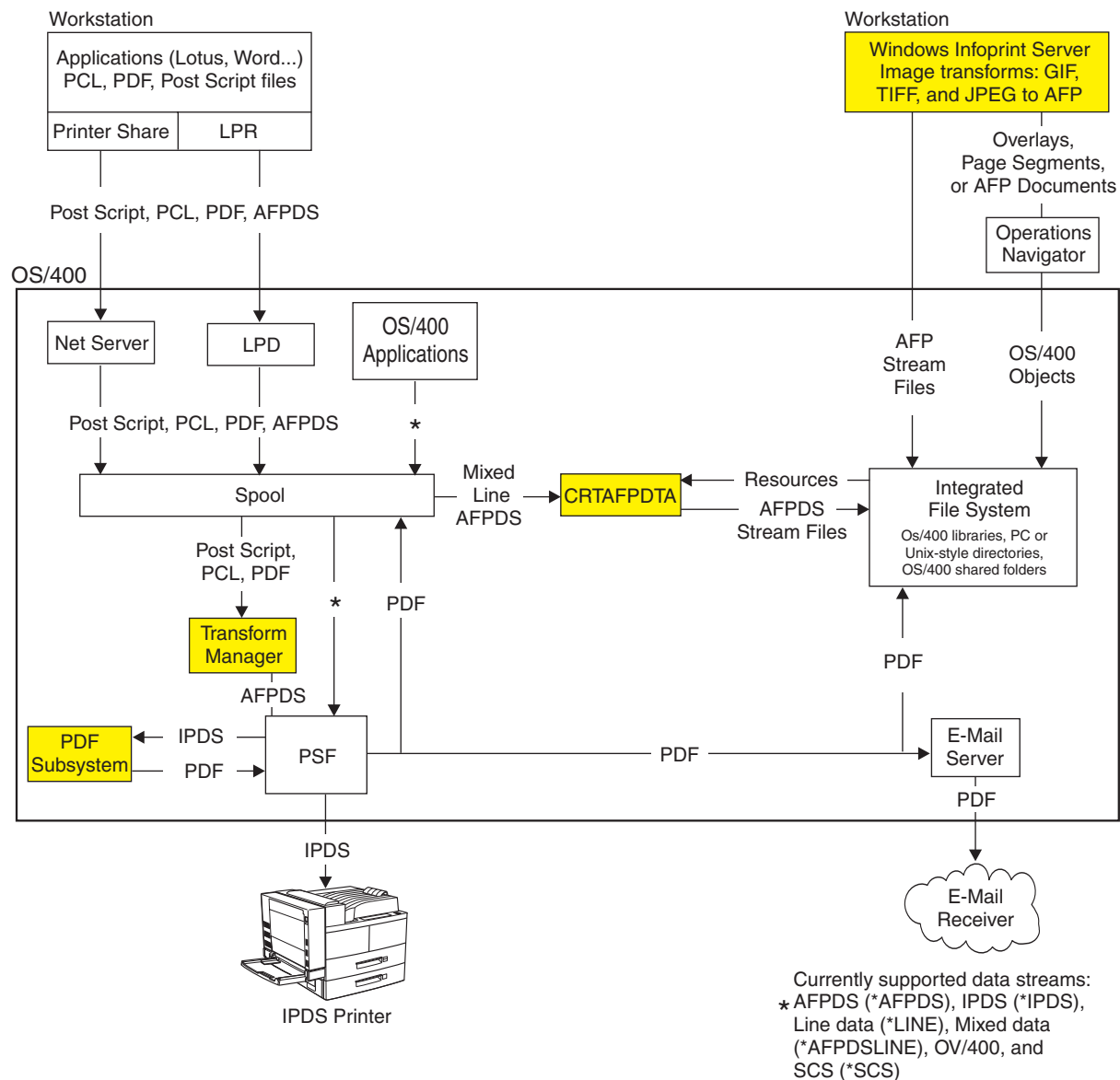


Figure 4. Infoprint Server Components

## What's New in Infoprint Server Version 5.2?

Infoprint Server 5.2 provides these enhancements:

- Use an SMTP server to send e-mail:

Previously, the only way to send e-mail from Infoprint Server was through the SNDDST command. With Infoprint Server 5.2, you can specify that Infoprint Server uses an SMTP server to send your e-mail.

- Individualize e-mails that are sent by the PDF subsystem:

With the prior version of Infoprint Server, you could only use a PDF mapping program to map mail tags when you specified a keyword, such as a customer number, for the mail tag. Now you can use the PDF mapping program to interpret mail tags, specify the subject text, and add a customized message to the beginning of each e-mail. If you use an SMTP mail server to send the e-mail, you

can also specify a file to be used in the body of the e-mail, add carbon copy (cc) and blind carbon copy (bcc) recipients, specify an address for the recipient to reply to, and add attachments.

- Use fonts from your Mac when transforming PostScript data to AFP:  
With Infoprint Manager for Windows or Infoprint Manager for AIX® along with Infoprint Server Font Downloader, you can upload DBCS fonts from your Mac to the iSeries. You can then use these fonts with the PostScript to AFP transform.
- Transform enhancements:  
The PDF, PCL, and PostScript to AFP transforms that are shipped with Infoprint Server 5.2 have been enhanced for better performance. The PCL to AFP transform now supports PCL 6.
- Indexing for PDF files:  
When transforming a file to PDF, Infoprint Server can now place index tags at group boundaries and return one PDF file. This lets you easily navigate the file when viewing it on your workstation.
- Smaller PDF files:  
Infoprint Server 5.2 lets you generate a PDF file without the fonts embedded. This lets you produce smaller PDF files.
- Enhancements to Create AFP Data:  
Easier file management: Create AFP Data (CRTAFPDTA) now lets you tell Infoprint Server to delete the output stream files after it merges them together. Two new parameters make it easier to identify the input file.
- Interactive access to PDF and e-mail functions  
Infoprint Server for iSeries, together with iSeries Access 5.2, lets you use the fully graphical iSeries command interface to run PDF and e-mail functions interactively. Send output by e-mail in one step. Write single or multiple output files to the integrated file system.
- Support printing with new iSeries Web Access  
Infoprint Server for iSeries, when installed with iSeries Access, enables direct PDF printing from your browser.

---

## Installing Infoprint Server

Refer to *Software Installation*, SC41-5120-06 for information about installing iSeries programs.

---

## Chapter 2. Using the Create AFP Data (CRTAFPDTA) Command

This chapter describes how to use the Create AFP Data (CRTAFPDTA) command. It contains these sections:

- “Overview of CRTAFPDTA” describes what the command can do and how it works.
- “Syntax” on page 9 gives a syntax diagram of the command.
- “Planning for CRTAFPDTA” on page 11 describes what you need to do and know before using CRTAFPDTA.
- “Indexing with CRTAFPDTA” on page 12 describes how to use CRTAFPDTA to index a spooled file, lists indexing limitations, and describes how to use AFP Viewer to view the output file.
- “Retrieving Resources” on page 14 describes how CRTAFPDTA gathers the resources that are needed to print or view the document.
- “CRTAFPDTA Parameters” on page 15 describes the parameters in detail.
- “CRTAFPDTA Examples” on page 25 gives examples of using the CRTAFPDTA command. It shows how to merge an AFP file with its resources and how to index a file.
- “E-mailing the CRTAFPDTA Output” on page 32 describes how to use the PDF subsystem to e-mail the output from CRTAFPDTA.
- CRTAFPDTA implements functionality available on other platforms with the tool AFP Conversion and Indexing Facility (ACIF). “CRTAFPDTA and ACIF” on page 34 gives two tables that cross reference CRTAFPDTA parameters with ACIF parameters.

---

### Overview of CRTAFPDTA

CRTAFPDTA transforms line (\*LINE) or mixed (\*AFPDSL) data into AFP (\*AFPDS) data. It can also take an AFP file as input and output an AFP stream file. It indexes a document for viewing, archiving, or document retrieval, and optionally retrieves and packages the AFP resources that are needed for printing or viewing.

CRTAFPDTA takes input data from the spool and creates up to four stream files in the integrated file system:

#### **AFP stream file**

The AFP document. This file is always created. Specify a value for **To stream file** (TOSTMF) to generate this file.

#### **Resource stream file**

A stream file that contains all the resources needed for printing or viewing. This file is optional, but is useful for archiving and viewing the document. Specify a value for **To resource stream file** (TORSCSTMF) to generate this file.

#### **Index stream file**

A stream file that contains indexing information. This file is optional, but is useful for viewing the document. Specify a value for **To index stream file** (TOIDXSTMF) to generate this file.

#### **Merged stream file**

A stream file that merges the AFP file with any of the other above files that are

created. This file is optional, but is useful for archiving and viewing the document. Specify a value for **To merged stream file (TOMRGSTMF)** to generate this file.

After these files are created, you can use them in an archive system, send them for use on another system, view them with AFP Viewer, use them on the Web, or send them to the PDF Subsystem to be made into one or more PDF files. Figure 5 illustrates this process:

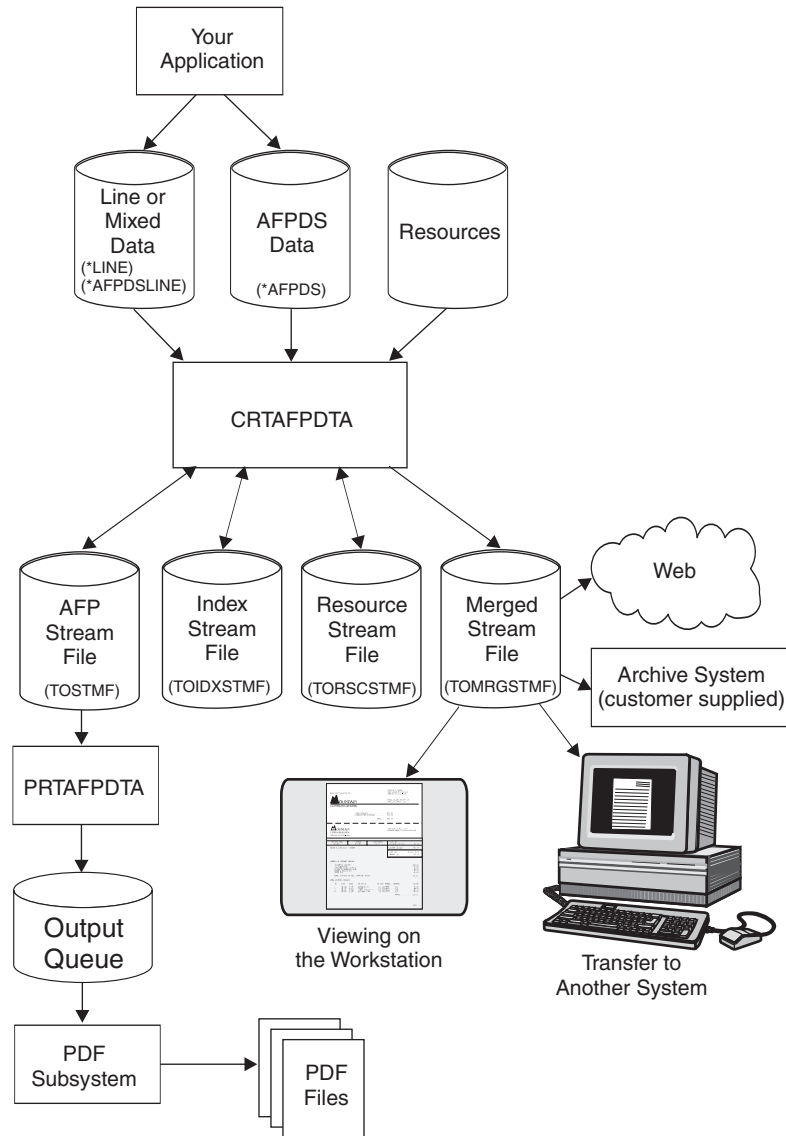


Figure 5. Using CRTAFPDTA

If you archive the resources along with the document, you can print with fidelity forever, even if the resources have changed since the document was converted. For example, suppose that a page segment contains a company officer's signature and is included in the print data. When someone new replaces the officer, current print files need to reference the new officer's signature. However, archived files must reference the former officer's signature.



The input to this command is a spooled file. The spooled file can be line, mixed, or AFPDS.

**Note:** These values are obtained from the spooled file attributes and cannot be specified on CRTAFPDTA:

- Control character
- Table reference characters
- DBCS SO/SI spacing

The output from this command is one or more stream files in directories.

**Note:** In this chapter whenever the term "file" is used by itself, it means "stream file".

CRTAFPDTA implements functionality available to other platforms with ACIF. ACIF is part of PSF on MVS, VM, and VSE and is also part of Infoprint Manager on AIX and Windows NT. Table 1 on page 34 gives the ACIF equivalencies to CRTAFPDTA parameters. Table 2 on page 35 gives the CRTAFPDTA equivalencies to ACIF parameters. You can find more information about using the functions available through the CRTAFPDTA command in *AFP Conversion and Indexing Facility*. User's Guide, S544-5285.

---

## Syntax

Figure 6 on page 10 shows the command syntax of CRTAFPDTA. For information about the parameters, see "CRTAFPDTA Parameters" on page 15.

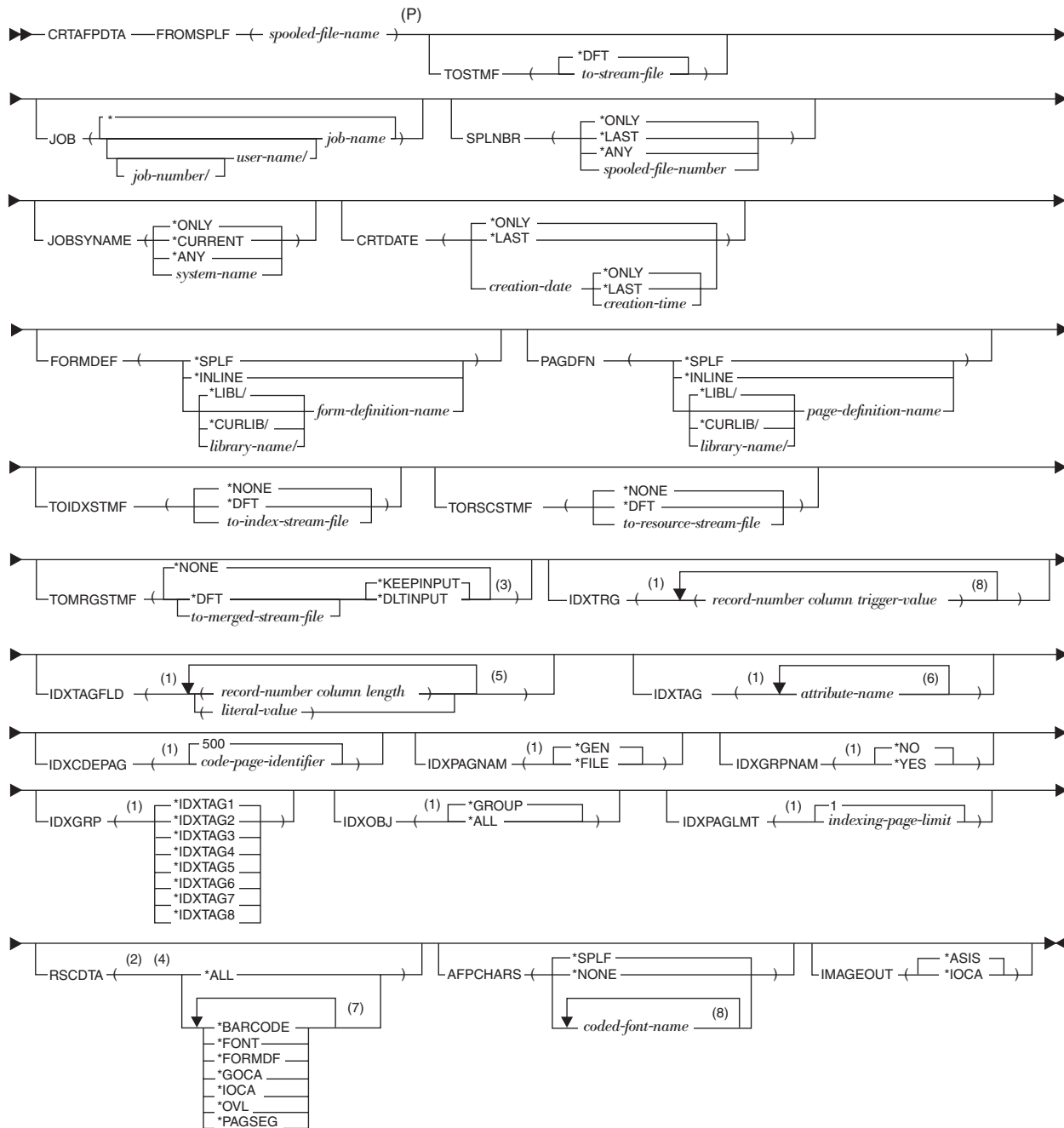


Figure 6. CRTAFPDTA command syntax

#### Notes:

1. Parameter only valid if TOIDXSTMF value is not \*NONE.
2. Parameter only valid if TORSCSTMF value is not \*NONE.
3. Parameter only valid if either TOIDXSTMF value or TORSCSTMF value is not \*NONE.
4. One or more resource data types must be specified if TORSCSTMF value is not \*NONE.
5. A maximum of 16 repetitions.
6. A maximum of eight repetitions.
7. A maximum of seven repetitions.

- 8. A maximum of four repetitions.
- P. All parameters preceding this point can be specified in positional form.

---

## Planning for CRTAFPDTA

This section describes what you need to do before using CRTAFPDTA and lists the authorities you need in order to use the command.

Before using CRTAFPDTA, you need to decide what you want done with the input file.

- If you want to index a file, you must decide whether you are going to index with data values or with literal values. You also need to decide which values to use. For information about indexing a file, see “Indexing with CRTAFPDTA” on page 12.
- You need to decide whether you want to view the final output. For information about viewing the output see “Viewing the Indexed Output” on page 31.
- You need to decide whether you will want to print the exact same document at a later date. For information about retrieving resources so you can print the same document later, see “Retrieving Resources” on page 14.

## Authorities

To use CRTAFPDTA, you must be authorized to all resources and files needed by the command.

### Spooled File

You must be authorized to the input spooled file by meeting at least one of these conditions:

- You own the input spooled file.
- You have \*READ authority to the spooled file’s queue, the queue is specified as DSPDTA(\*YES), and you have \*EXECUTE authority to the queue’s library.
- You have \*SPLCTL special authority.
- You have \*SPLNBRCTL special authority, the spooled file’s queue is specified as OPRCTL(\*YES), and you have \*EXECUTE authority to the queue’s library.
- You own the spooled file’s queue, the queue is specified as AUTCHK(\*OWNER), and you have \*EXECUTE authority to the queue’s library.
- You have read, add, and delete authority to the spooled file’s queue and the queue is specified as AUTCHK(\*DTAAUT). You also have \*EXECUTE authority to the queue’s library.

### AFP Resources

You must have all of these authorities to all processed AFP resources:

- **Resource Authority:** \*USE
- **Library Authority:** \*EXECUTE
- **Resource Lock:** \*SHRRD

### Input and Output Stream Files

You must have all of these authorities to all the stream files you use:

- **Authority to directories in the path where the stream file resides:** \*X
- **Authority to merged stream file:** \*W
- **Authority to stream files that are input to merged stream file:** \*R
- **Authority to parent directories of new stream files:** \*WX

---

## Indexing with CRTAFPDTA

Indexing is one of the primary tasks for which CRTAFPDTA is used. CRTAFPDTA indexes a document by inserting tags in the print file and, optionally, in the index object file. Indexing lets you divide a large print file into uniquely identified smaller units, called *groups*. A group is a named collection of sequential pages. For example, you can divide a large bank statement application into groups that are made up of individual statements. Each statement is uniquely identified by its group name, which might be the account number or other attributes such as the date and a Social Security number.

One of the main reasons to use CRTAFPDTA to index a file is for use with the PDF subsystem. Once indexed, the PDF subsystem can take the file, create a PDF file from each group, and e-mail each PDF file to a different person. In the example above, you could send the indexed bank statement output to the PDF subsystem then e-mail each statement to the appropriate customer.

Using CRTAFPDTA, you can also create an index object file, which lets you do this:

- Retrieve individual statements from storage, based on an account number or any other attribute.
- More rapidly access the statements for viewing with, for example, AFP Viewer.
- Archive individual statements or the entire indexed print file for long-term storage and subsequent data management and reprinting, even years after its creation.

Figure 7 shows the relationship between the group-level tags and the entries in the index object file.

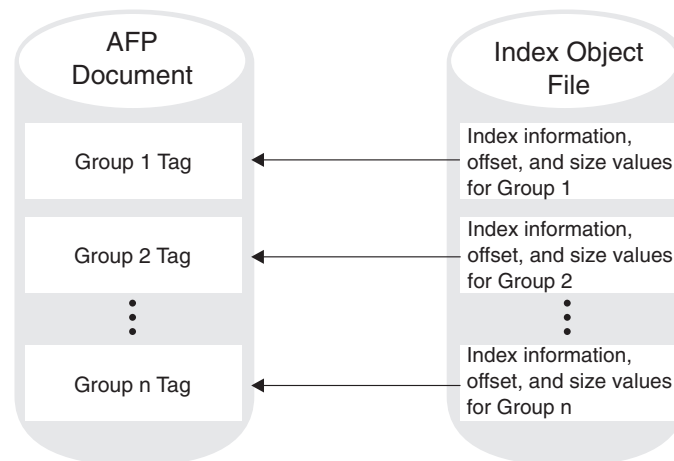


Figure 7. AFP Document with Index Tags and the Index Object File

CRTAFPDTA lets you generate the indexing tags in one of these ways:

- Using values present in the input data. This is useful when the data has been formatted so that CRTAFPDTA can reliably locate the values. This kind of indexing is called *indexing with data values*.
- Using literal values that you specify to CRTAFPDTA. This is useful when the values you want to use in the indexing tags are not consistently present in the data. This kind of indexing is called *indexing with literal values*.

## Indexing with Data Values

When you index with data values, you use data that is in the same place in every group to identify a group of pages. If you index a bank statement, the account number is a type of data value that you might want to use. You can then use the account number to archive a single customer's account statement. This lets you use the account number to retrieve and view the statement. This type of indexing uses the **Index trigger definition (IDXTRG)**, **Index field definition (IDXTAGFLD)**, and **Index tag definition (IDXTAG)** parameters. Use **To merged stream file (TOMRGSTMF)** to generate a merged stream file if you want to view the output file with indexing and resources. See "Indexing with CRTAFPDTA" on page 26 for an example of indexing with data values.

## Indexing with Literal Values

Some print files, such as technical documents and memos, do not have a data value is consistently present in the same location. Therefore, CRTAFPDTA cannot easily use data values to divide the document into groups of pages. Likewise, the output of an application might not contain the data you want to use for an indexing tag. In these cases, you can specify one or more literal values for CRTAFPDTA to use as indexing tags for a single group of pages. This type of indexing uses the **Index field definition (IDXTAGFLD)** parameter. Use **To merged stream file (TOMRGSTMF)** to generate a merged stream file if you want to view the output file with indexing and resources. In "Indexing with CRTAFPDTA" on page 26, the literal value Telephone Bill is used.

### Notes:

1. If try to use CRTAFPDTA to add indexing tags to a file that already contains indexing tags, CRTAFPDTA issues an error message and stops processing. If the input file already contains indexing tags, you can create the index object file by running CRTAFPDTA without specifying any indexing parameters.
2. CRTAFPDTA includes the name of the output document in the index object file and includes the name of the index object file in the output document. This gives a method of correlating the index object file with the appropriate output document.

## Indexing Limitations

If you have a line-mode application that is not suitable for indexing with data values or literal values, the application program cannot insert tagging structured fields in the print data. This is because tagging structured fields are not allowed in line data. To solve this problem, you can place index values in a No Operation (NOP) structured field. Then use those values with appropriate values for **Index trigger definition (IDXTRG)** and **Index field definition (IDXTAGFLD)** to index the file. For information about structured fields, refer to *Data Stream and Object Architectures: Mixed Object Document Content Architecture Reference*, SC31–6802.

When application data does not contain appropriate data values for indexing, the application can add the index triggers. One possible location is the record containing the new-page carriage control character (for example, a skip-to-channel 1). To do this, the application needs to add the indexing trigger and attribute value to this record at a specified location on each statement in the print file. This lets CRTAFPDTA retrieve the information at processing time.

---

## Retrieving Resources

CRTAFPDTA can determine the list of AFP resources, such as fonts and page segments, that are needed to view or print the document. It can then retrieve these resources and store them in a stream file. CRTAFPDTA can then merge these resources with the AFP output file. This function is especially valuable if the resources are not present on the designated platform in a distributed print environment. If you merge the resources with the AFP output file, you can archive the document in the current form. This allows you to reproduce the document with fidelity at a later date, even if the resources have changed. Merging an AFP file with its resources also lets you use the AFP file with a Web browser or send it to other systems without worrying about resource availability.

The type of resources CRTAFPDTA retrieves from specified libraries is based on the value of the **Resource Data** (RSCDTA) parameter. When CRTAFPDTA processes a print file, it does this:

- Identifies the resources requested by the print file  
While CRTAFPDTA converts the input file to AFP, it builds a list of all the resources necessary to print the document. This includes all the resources referenced inside other resources. For example, a page can include an overlay and an overlay can reference other resources such as fonts and page segments.
- Creates a resource file  
CRTAFPDTA creates a logical resource library in the form of an AFP resource group and stores this resource group in a resource file. Specifying **RSCDTA(\*BARCODE, \*FONT, \*FORMDF, \*GOCA, \*IOCA, \*OVL, \*PAGSEG)** or **RSCDTA(\*ALL)** guarantees that the resource file contains all the resources necessary to view or print the document with fidelity.
- Includes the name of the output document in the resource file and the name of the resource file in the output document  
This provides a method of correlating resource files with the appropriate output document.

An example of specifying CRTAFPDTA parameters for resource retrieval can be found in “CRTAFPDTA Examples” on page 25.

## How CRTAFPDTA Searches for AFP Resources

CRTAFPDTA looks for resources in several different places. You can library-qualify a resource if you specify it in one of these ways:

- Use the **Form definition** (FORMDF) or **Page definition** (PAGDFN) parameters to specify a form definition or page definition directly to CRTAFPDTA.
- Specify a resource in the input spooled file's attributes.
- Use DDS keywords, such as OVERLAY or PAGSEG to specify a resource.

The library-qualifier can be \*LIBL, \*CURLIB, or a library name. If a resource is not library-qualified, CRTAFPDTA searches \*LIBL for the resource. If \*LIBL or \*CURLIB is used as the library-qualifier and a resource which you are authorized to use is not found, CRTAFPDTA searches the font libraries QFNTCPL, QFNT01-QFNT19, and QFNT61-QFNT69 for the resource. If the resource is still not found, CRTAFPDTA issues an error message and stops processing.

---

## CRTAFPDTA Parameters

This section describes each parameter used with the CRTAFPDTA command. The parameters are listed alphabetically by keyword.

### AFP characters (AFPCHARS)

Specifies one or more AFP coded fonts to use with line data and a page definition.

**\*SPLF** Use the values for **AFP characters** (AFPCHARS) that are specified in the spooled file attributes for the input spooled file.

**\*NONE**

No AFP characters are specified.

#### ***coded-font-name***

Specifies up to four 4-byte names. The 4-byte names are concatenated to 'X0' to identify up to four coded fonts which are used when table reference characters (spooled file attribute TBLREFCHR) are in the data.

### Spooled file created (CRTDATE)

Specifies the date and time the input spooled file was created.

**\*ONLY** There is only one spooled file with the specified job name, user name, job number, spooled file name, spooled file number, and job system name.

**\*LAST** Use the spooled file created most recently with the specified job name, user name, job number, spooled file name, spooled file number, and job system name.

#### ***spooled-file-create-date***

Specify the date the spooled file was created. If you specify *spooled-file-create-date*, you must also specify a value for the time the spooled file was created.

The possible time values are:

**\*ONLY**

There is only one spooled file with the specified job name, user name, job number, spooled file name, spooled file number, job system name, and spooled file creation date.

**\*LAST** Use the spooled file created most recently with the specified job name, user name, job number, spooled file name, spooled file number, job system name, and spooled file creation date.

#### ***spooled-file-create-time***

Specify the time the spooled file was created.

### Form definition (FORMDF)

Specifies the form definition to use when processing the spooled file. A form definition is a resource that defines the characteristics of the physical form. Some of these characteristics are overlays to be used, the position of page data on the form, number of copies of pages, and modification to pages.

**\*SPLF** Use the value specified for **Form definition** in the spooled file's attributes. CRTAFPDTA cannot use \*DEVD from the spooled file's **Form**

**definition** attribute. If \*DEVD is specified, either change the spooled file attribute by naming a form definition or by using *form-definition-name* to specify a form definition for CRTAFPDTA.

**\*INLINE**

A form definition is included in the input data.

***form-definition-name***

Specifies the name of the form definition. A form definition name can be up to eight characters long. This form definition overrides any specified in the spooled file's attributes or inline in the data. You must also specify the library that contains the form definition.

The possible library values are:

**\*LIBL** All libraries in the job's library list are searched until the form definition is found. If the form definition is not found, the font libraries QFNTCPL, QFNT01-QFNT19, and QFNT61-QFNT69 are searched.

**\*CURLIB** The form definition is in the current library for the job. If no library is specified as the current library for the job, QGPL is used. If the form definition is not found, the font libraries QFNTCPL, QFNT01-QFNT19, and QFNT61-QFNT69 are searched.

***library-name*** Specifies the library where the form definition is located. The form definition must be in the specified library.

**From spooled file (FROMSPLF)**

This parameter is required. It specifies the name of the spooled file that contains the data that CRTAFPDTA processes.

***spooled-file-name***

Specifies the name of the spooled file that CRTAFPDTA processes. CRTAFPDTA can process \*LINE, \*AFPDLINE, or \*AFPDS data.

**Index code page identifier (IDXCDEPAG)**

Specifies the code page to be used for the index values and attribute names produced by CRTAFPDTA. This value is used by programs that display indexing information. These programs use this value with code page translation tables to represent the information.

This parameter is only valid if **To index stream file (TOIDXSTMF)** has a value other than \*NONE.

**500** Code page identifier 500.

***code-page-identifier***

A code page identifier from 1–65535.

**Select index for group names (IDXGRP)**

Specifies which of the eight possible **Index tag definition (IDXTAG)** values should be used as the group name for each index group. IBM recommends that you use the most unique index value for the group name. The intent is for each group in the output file to have a unique name. For instance, assume you used a customer's name, phone number, area code, and the date to index phone bills, as in "Indexing with CRTAFPDTA" on page 26. The most unique index value is the customer name or phone number. The value for this parameter



uses the **Index field definition** (IDXTAGFLD) definitions from the **Index tag definition** parameter but does not include the attribute name.

The AFP Viewer displays this value along with the attribute name and index value. You can use the group name to select a group of pages to view. For more information about viewing indexed documents, refer to *AS/400 Guide to Advanced Function Presentation and Print Services Facility*.

This parameter is only valid if **To index stream file** (TOIDXSTMF) has a value other than \*NONE.

**\*IDXTAG1**

Use the first **Index tag definition** value.

**\*IDXTAG2-\*IDXTAG8**

Use the specified **Index tag definition** value.

**Generate group names (IDXGRPNAM)**

Specifies whether CRTAFPDTA creates a unique group name for each group in the output file.

This parameter is only valid if **To index stream file** (TOIDXSTMF) has a value other than \*NONE.

**\*YES** CRTAFPDTA creates a unique group name by generating an 8-character numeric string and appending the string to the group name.

**\*NO** CRTAFPDTA does not generate the string. Specify \*NO if the group name is a mail tag.

**Information for index file (IDXOBJ)**

Specifies how much information CRTAFPDTA writes to the index object file. This determines the level of detail that appears in the index.

This parameter is only valid if **To index stream file** (TOIDXSTMF) has a value other than \*NONE.

**\*GROUP** Only group-level entries are put into the index object file.

**\*ALL** Both page-level and group-level entries are put into the index object file. Only specify this value if you have TLEs in the input datastream. For information about TLEs, refer to *AFP Toolbox for Multiple Operating Systems: User's Guide* or *DDS Reference*.

**Note:** The online help for this value is incorrect.

**Indexing page limit (IDXPAGLMT)**

Specifies the output page number by which CRTAFPDTA must find an indexing field if CRTAFPDTA is indexing the file. If CRTAFPDTA does not find an indexing field by the end of the specified page, processing stops.

This parameter is helpful if your file contains header pages. You can specify a page number one greater than the number of header pages, so that CRTAFPDTA will continue to look for matches after the header pages.

This parameter is only valid if **To index stream file** (TOIDXSTMF) has a value other than \*NONE.

**\*1** An indexing field must be found in the first output page.

***indexing-page-limit***

A number from 1–99.

## Generate page names (IDXPAGNAM)

Specifies whether CRTAFPDTA generates page names using an 8-byte counter or uses structured field tokens that are found in the input data stream. If the input data contains Begin Page structured fields (BPGs) with Fully Qualified Name triplets (FQNs), CRTAFPDTA does not generate page names.

Use of this parameter requires technical knowledge of the makeup of the input file. Such users will understand the explanation of this parameter. All other users should use the default value of \*GEN.

This parameter is only valid if **To index stream file** (TOIDXSTMF) has a value other than \*NONE.

**\*GEN** CRTAFPDTA uses an 8-byte counter to generate page names.

**\*FILE** CRTAFPDTA uses structured field tokens in the input data stream to generate page names.

## Index tag definition (IDXTAG)

Specifies the content of the indexing tags for the entire file. If literal values are specified for every index tag definition, CRTAFPDTA treats the entire file as one page group and uses this information to index the document. CRTAFPDTA stops processing the input file if literal values are specified for all **Index tag definition** parameters and any **Index trigger definition** (IDXTRG) parameters are also specified.

A maximum of eight attribute names can be defined, and each attribute name can be made up of more **Index field definition** (IDXTAGFLD) definitions.

This parameter is only valid if **To index stream file** (TOIDXSTMF) has a value other than \*NONE.

### ***attribute-name***

Specifies any combination of these:

#### ***literal-value***

Specifies a user-defined attribute name to be associated with the actual index value. For example, assume that the first index tag definition is a customer's bank account number. The value of **Index tag definition** is the account number (for example, 1234567), so the string 'account number' would be a meaningful attribute name.

The attribute name is an EBCDIC character string 1–250 bytes long. Think of it as a label for the actual index value. CRTAFPDTA does not perform any validity checking on the contents of the attribute name.

***field...*** Specifies up to 16 **Index field definition** (IDXTAGFLD) definitions that compose the index value such as \*IDXTAGFLD1 (specifies the first value for **Index field definition**). If more than one **Index trigger definition** (IDXTRG) definition is specified, CRTAFPDTA concatenates them into one physical string of data. No delimiters are used between the concatenated fields.

An **Index tag definition** (IDXTAG) value has a maximum length of 250 bytes. Therefore, the total length of all specified index trigger definitions for a single index cannot exceed this length. CRTAFPDTA reports an error condition and stops processing if **Index tag definition** is too long.

For **Index trigger definition** values that specify data values within the file, CRTAFPDTA determines the actual location of the indexing information based on the indexing anchor record. The indexing anchor record is defined by the **Index trigger definition** values.

**Note:** If one or more **Index trigger definition** parameters are specified (that is, CRTAFPDTA indexes the file), at least one **Index tag definition** parameter must be specified. Also, that index tag must contain at least one *field* parameter value that is not a literal. CRTAFPDTA reports an error condition and stops processing if this rule is not satisfied.

For identification, CRTAFPDTA numbers each index tag based on the order in which they are specified. For example, if a message indicates an error with respect to Index tag definition number 2, a reference is being made to the second index tag you specified. Specify index tag values by number (for example, \*IDXTAG1) for the **Select index for group names** (IDXGRP) parameter.

### **Index field definition (IDXTAGFLD)**

Specifies the data fields, such as the account numbers, used to construct the indexing information. These data fields can be specified as literal values or CRTAFPDTA can retrieve the data from the input records of the file. You can define at most 16 fields.

This parameter is only valid if **To index stream file** (TOIDXSTMF) has a value other than \*NONE.

#### ***record-number or literal-value***

Specifies the record number relative to the indexing anchor record, or a literal value.

##### ***record-number***

Specifies the record number relative to the indexing anchor record, which is the first index trigger definition. If you specify *record-number*, you must also specify the *column* and *length*. To specify a record number, enter the number without quotation marks ( ' ).

##### ***literal-value***

Specifies a literal (constant) value. To specify a literal value, enclose the value in quotation marks ( ' ). If you specify a literal value, any values provided for column and length are ignored.

#### ***column***

Specifies the byte offset from the beginning of the record to the data field, where 1 is the first byte in the record. For files that contain carriage control characters, column 1 refers to the carriage control character.

For applications that use specific carriage control characters to define page boundaries, for example, skip to channel 1, consider defining the value of the carriage control character as one of the **Index trigger definition** (IDXTRG) parameters.

If the specified value exceeds the physical length of the record, CRTAFPDTA reports an error condition and stops processing.

Specify a column number only when a record number has been specified.

*length* Specifies the number of contiguous bytes (characters), starting at the value specified for *column*, that make up this field. The field can extend outside the record length as long as the column where it begins is in the record length. In this case, CRTAFPDTA adds padding blanks (X'40') to fill out the record. If the field begins outside the length of the record, CRTAFPDTA reports an error condition and stops processing.

Specify a length only when a record number has been specified.

Each field is numbered automatically by CRTAFPDTA based on the order in which you specify the values. The first field you define is numbered one. For example, if a message indicates an error with respect to Index field definition number 9 (\*IDXTAGFLD9), it is referencing the ninth Index field definition you specified.

The **Index tag definition** (IDXTAG) parameter allows you to use index field definitions as part of an index tag definition. For example, to refer to the third index field definition and label that field NAME when specifying an index tag definition, specify IDXTAG(('NAME' (\*IDXTAGFLD3))).

### Index trigger definition (IDXTRG)

Specifies the locations and values of data strings in the input file that are used to define indexing groups. Each **Index trigger definition** specification is made up of three values, the *record-number*, *column-number*, and *value*. When CRTAFPDTA finds *value* in column number *column-number* of record *record-number*, it starts a new page group.

These data strings are referred to as triggers because their presence in the file triggers a processing action. The first **Index trigger definition** specification is special. Each record in the file containing the value specified in the first **Index trigger definition** specification is referred to as an *indexing anchor record*. The indexing anchor record is the reference point from which all indexing information is found.

The number of **Index trigger definition** specifications required to uniquely identify the beginning of a group of pages depends on the complexity of the application output. You can specify at most four triggers.

This parameter is only valid if **To index stream file** (TOIDXSTMF) has a value other than \*NONE.

#### ***record-number or \****

Specifies the relative record number from the indexing anchor record.

#### ***record-number***

A number between 0-255.

- \* CRTAFPDTA checks every record for the first **Index trigger definition** specification. For example, if you specify IDXTRG(\*,2,'1'), CRTAFPDTA looks in every record until it finds the number 1 in column two. You must specify an asterisk ( \* ) in the first **Index trigger definition** specification. You cannot specify an asterisk in any other **Index trigger definition** specification.

#### ***column-number or \****

Specifies the byte offset from the beginning of the record to the trigger value. The possible values for column-number are:

**column-number**

A value from 1 to 32756, where 1 is the first byte in the record. For files that contain carriage control characters, column 1 refers to the carriage control character. CRTAFPDTA compares *trigger-value* to the input data. If the specified trigger value exceeds the physical length of the record, CRTAFPDTA considers the comparison false and continues processing.

- \* CRTAFPDTA scans the record from left to right, looking for the trigger-value.

**trigger-value**

Specifies the actual alphanumeric or hexadecimal value of the trigger. Surround the value with quotation marks ( ' ). This value is case sensitive and cannot be more than 250 bytes long.

CRTAFPDTA does not perform any validity checking on this value, but uses it to perform a byte-for-byte comparison with the records in the file. If the combined values of *column-number* and the *trigger-value* length exceed the physical length of the record, CRTAFPDTA considers the comparison false and continues processing.

CRTAFPDTA numbers each trigger based on the order in which they are specified. For example, if a message indicates an error with respect to Index trigger definition number 4 (IDXTRG4), a reference is being made to the fourth Index trigger definition you specified.

**Image output (IMAGEOUT)**

Specifies the format of the image data produced by CRTAFPDTA in the output document.

- \*ASIS** The format of the image data is not changed.
- \*IOCA** All image data is output in uncompressed IOCA.

**Job name (JOB)**

Specifies the name of the job that created the input spooled file.

- \* The job that issued this command created the spooled file.
- job-name** The name of the job that created the spooled file.
- user-name** The user name that identifies the user profile under which the job was run.
- job-number** The system assigned job number.

**System name (JOBSYSNAME)**

Specifies the name of the system where the job that created the input spooled file (specified on the **Job name** (JOB) parameter) was run.

- \*ONLY** There is only one spooled file with the specified job name, user name, job number, spooled file name, spooled file number, and spooled file creation date and time.
- \*CURRENT** Use the spooled file created on the current system with the specified job name, user name, job number, spooled file name, spooled file number, and spooled file creation date and time.
- \*ANY** Do not consider the job system name when selecting a spooled file. Use this value when you want the **Spooled file create**

**date** and **Spooled file create time** values to take precedence over the job system name when selecting a spooled file.

***job-system-name***

Specify the name of the system where the job that created the spooled file ran.

**Page definition (PAGDFN)**

Specifies the page definition used to format line data. This parameter is only used if the input data stream is line (\*LINE) or mixed mode (\*AFPDLINE). You can use a page definition shipped with OS/400, create one with a utility such as PPFA, or use one created by a service. For the list of page definitions shipped with OS/400 refer to *Printer Device Programming*. For information about creating page definitions, see *Page Printer Formatting Aid: User's Guide*.

**\*SPLF** Use the value specified for **Page definition** in the spooled file's attributes. CRTAFPDTA cannot use \*DEVDT from the spooled file's **Page definition** attribute. If \*DEVDT is specified, either change the spooled file attribute by naming a page definition or by using *page-definition-name* to specify a page definition for CRTAFPDTA.

**\*INLINE**

A page definition is included in the input data.

***page-definition-name***

Specifies the name of the page definition. A page definition name can be up to eight characters long. You must also specify the library that contains the page definition. This page definition overrides any specified in the spooled file's attributes or inline in the data.

**The possible library values are:**

**\*LIBL** All libraries in the job's library list are searched until the page definition is found. If the page definition is not found, the font libraries QFNTCP, QFNT01-QFNT19, and QFNT61-QFNT69 are searched.

**\*CURLIB**

The page definition is in the current library for the job. If no library is specified as the current library for the job, QGPL is used. If the page definition is not found, the font libraries QFNTCP, QFNT01-QFNT19, and QFNT61-QFNT69 are searched.

***library-name***

Specifies the library where the page definition is located. The page definition must be in the specified library.

**Resource Data (RSCDTA)**

Specifies the types of resource data that CRTAFPDTA writes to the resource stream file. Your library list (\*LIBL) is searched for the needed resources. Specify **\*ALL** or up to seven types of resources.

This parameter is only valid if **To resource stream file** (TORSCSTMF) has a value other than \*NONE. One or more resource data types must be specified if **To resource stream file** has a value other than \*NONE.

**\*ALL** All resources required to print or view the output document (specified on the **To stream file** parameter) are included in the resource stream file. When **\*ALL** is specified, you cannot specify other resource types.

**\*BARCODE** Specifies that all BCOCA objects included by an Include Object

(IOB) structured field required to print or view the output document are included in the resource stream file.

**\*FONT** Specifies that all font character sets, code pages, and coded fonts required to print or view the output file are included in the resource stream file.

If you only want to view the file, do not specify that fonts are to be included in the resource stream file. The AFP Viewer does not use these fonts; it substitutes fonts that are on the PC running AFP Viewer. If you do not include fonts in the resource stream file, you can minimize the size of the stream file.

**\*FORMDF** Specifies that the form definition used in processing the file is included in the resource stream file.

**\*GOCA** Specifies that all GOCA objects included by an IOB structured field required to print or view the output document file are included in the resource stream file.

**\*IOCA** Specifies that all IOCA objects included by an IOB structured field required to print or view the output document file are included in the resource stream file.

**\*OVL** Specifies that all overlays required to print or view the output document file are included in the resource stream file.

**\*PAGSEG** Specifies that all page segments required to print or view the output document file are included in the resource stream file.

#### Spooled file number (SPLNBR)

Specifies the number of the input spooled file from the job specified for **Job name** (JOB).

##### \*ONLY

Only one spooled file from the job has the name specified in **From spooled file** (FROMSPLF). If you specify \*ONLY and more than one spooled file has the specified name, CRTAFPDAT issues an error message.

**\*LAST** Use the highest numbered spooled file with the specified name.

**\*ANY** Do not consider the file number when selecting a spooled file.

##### ***spooled-file-number***

Specifies the number of the spooled file indicated in **From spooled file**.

#### To index stream file (TIDXSTMF)

Specifies the stream file to which the index object file is written.

##### \*NONE

The index object information is not written to a stream file.

**\*DFT** The index object file is written to a file with a default name in the current working directory. If the current working directory references the QSYS.LIB file system, the file name is **outputidx.mbr**. Otherwise, the file name is **output.idx**.

##### *to-index-stream-file*

Specifies the path of the stream file to which the index object data is written. If the path name is qualified, it must be enclosed in quotation



marks. The path name cannot contain a pattern. All directories in the path must exist. New directories are not created. If the stream file does not exist, it is created.

If you specify the file name without a path, the output file is placed in your current working directory. If the file exists, it is overwritten. If you name a directory or path but not a file name, a file with a default name is written. If the specified directory references the QSYS.LIB file system, the file name is **outputidx.mbr**. Otherwise, the file name is **output.idx**.

### To merged stream file (TOMRGSTMF)

Specifies the stream file into which these stream files are copied:

- AFP stream file, specified with **To stream file** (TOSTMF)
- Index stream file, if it exists. The index stream file is specified with **To index stream file**.
- Resource stream file, if it exists. The resource stream file is specified with **To resource stream file** (TORSCSTMF).

The original stream files remain on your system unless \*DLTINPUT is specified. Specify a value other than \*NONE for **To merged stream file** if the document references resources that do not exist on all systems and you want to view the output, save the exact document for archival purposes, or send the document to another system. For information about viewing the merged file, refer to *AS/400 Guide to Advanced Function Presentation and Print Services Facility*.

#### \*NONE

The stream files are not merged.

If you want a merged stream file, use one of these parameters to specify its name:

**\*DFT** The stream files are written to a file with a default name in the current working directory. If the current working directory references the QSYS.LIB file system, the file name is **outputmrg.mbr**. Otherwise, the file name is **output.mrg**.

#### *to-merged-stream-file-path*

Specifies the path of the stream file to which the stream files are written. If the path name is qualified, it must be enclosed in quotation marks. The path name cannot contain a pattern. All directories in the path must exist. New directories are not created. If the stream file does not exist, the output file is placed in your current working directory.

If you specify the file name without a path, the output file is placed in your current working directory. If the file exists, it is overwritten. If you name a directory or path but not a file name, a file with a default name is written. If the specified directory references the QSYS.LIB file system, the default file name is **outputmrg.mbr**. Otherwise, the default file name is **output.mrg**.

If the merged file is created, specify whether to save the stream files that were copied into the merge file:

#### \*KEEPINPUT

Input stream files are not deleted.

#### **\*DLTINPUT**

The stream files used as input to the merged stream file are deleted after the merged stream file is written.



### To resource stream file (TORSCSTMF)

Specifies the stream file to which resources are written. One or more resource data types must be specified in **Resource Data** (RSCDTA) if **To resource stream file** has a value other than \*NONE.

#### \*NONE

Resources are not written to a stream file.

**\*DFT** The resource information is written to a file with a default name in the current working directory. If the current working directory references the QSYS.LIB file system, the file name is **outputrsc.mbr**. Otherwise, the file name is **output.rsc**.

#### *to-resource-stream-file*

Specifies the path of the stream file to which the resource data is written. If the path name is qualified, it must be enclosed in quotation marks. The path name cannot contain a pattern. All directories in the path must exist. New directories are not created. If the stream file does not exist, it is created.

If you specify the file name without a path, the output file is placed in your current working directory. If the file exists, it is overwritten. If you name a directory or path but not a file name, a file with a default name is written. If the specified directory references the QSYS.LIB file system, the file name is **outputrsc.mbr**. Otherwise, the file name is **output.rsc**.

### To stream file (TOSTMF)

Specifies the stream file to which CRTAFPDTA writes the transformed data.

**\*DFT** The transformed data is written to a file with a default name in the current working directory. If the current working directory references the QSYS.LIB file system, the file name is **outputafp.mbr**. Otherwise, the file name is **output.afp**.

#### *to-stream-file-path*

Specifies the path name of the stream file to which transformed data is written. If the path name is qualified, it must be enclosed in quotation marks. It cannot contain a pattern. All directories in the path name must exist. New directories are not created. If the stream file does not exist, it is created.

If you specify the file name without a path, the output file is placed in your current working directory. If the file exists, it is overwritten. If you name a directory or path but not a file name, a file with a default name is written. If the specified path references the QSYS.LIB file system, the file name is **outputafp.mbr**. Otherwise, the file name is **output.afp**.

---

## CRTAFPDTA Examples

This section shows the CRTAFPDTA command used these ways:

1. “Merging Resources with Data” on page 26 shows CRTAFPDTA being used to package resources with the data for viewing, archiving, or use on another system.
2. “Indexing with CRTAFPDTA” on page 26 shows a file being indexed with data and literal values, then merged for viewing, archiving, or use on another system.

See “Example” on page 33 for an example of a file being indexed and sent to the PDF subsystem for e-mailing.

## Merging Resources with Data

In this example, the AFP file MERGE\_ME is going to be archived. To ensure that we can print the document later exactly as it appears now, it is merged with the resources required to print or view it.

```
CRTAFPDTA
FROMSPLF(MERGE_ME)
TOSTMF(*DFT)
FORMDF(*SPLF)
PAGDFN(*SPLF)
TORSCSTMF(*DFT)
RSCDTA(*ALL)
TOMRGSTMF('/MYOUTPUT/MERGED.MRG' *DLTINPUT)
```

*Figure 8. Example CRTAFPDTA Command*

Figure 8 specifies this information:

**FROMSPLF(MERGE\_ME)**

MERGE\_ME is the input spooled file name.

**TOSTMF(\*DFT)**

Store the output AFP file in the default location with the default name.

**FORMDF(\*SPLF)**

Use the form definition specified in the spooled file attributes to process the input file.

**PAGDFN(\*SPLF)**

Use the page definition specified in the spooled file attributes to process the input file.

**TORSCSTMF(\*DFT)**

Store all of the resources needed to view and print the file in a stream file with the default name in the default location. The types of resources to store in this file are specified in **Resource Data** (RSCDTA).

**RSCDTA(\*ALL)**

Copy all resources (the form definition, page segments, overlays, fonts, bar codes, IOCA objects, and GOCA objects) that are needed to process this file into the resource stream file.

**TOMRGSTMF('/MYOUTPUT/MERGED.MRG' \*DLTINPUT)**

Merge the AFP file (TOSTMF) and the resource file (TOMRGSTMF) together. Name the new file MERGED.MRG and store it in the directory MYOUTPUT, then delete the AFP file and the resource file.

Default values are used for the parameters that are not listed above.

## Indexing with CRTAFPDTA

In this example, an application generates telephone bills, as is shown in Figure 9 on page 28. It then spools the data without specifying a page definition or form definition. The objective is to make the billing application output available on customer service representatives' workstations. Then, when a customer calls with a billing inquiry the representative can view the bill in the same format as a the customer's printed copy.


To achieve this objective, you must convert the output from the application into a document format that can be used with AFP Viewer. You also need to index the file

to facilitate searching the file with AFP Viewer. To ensure that all resources used in the bills are available at the workstation, you need to retrieve all resources.

These are the tasks:

- Examine the input file to determine how to tag it for indexing.
- Verify that your document correctly identifies the libraries in which referenced AFP resources reside. If \*LIBL is specified, confirm that the job running CRTAFPDTA has access to the necessary libraries. Also verify that the job running CRTAFPDTA has the required authority to use the referenced AFP resources.
- Use CRTAFPDTA parameters to specify the page definition and form definition, index the document, retrieve resources, and create a merged file.
- Use AFP Viewer to display the merged stream file.

For further description of any parameter used in this section, see “CRTAFPDTA Parameters” on page 15.



**BIG BEND**  
**TELEPHONE COMPANY**  
Serving Texans since 1903

Amount Due
\$ 242.11

Billy Bob Smith (214) 555-1234  
RFD 23  
Big Bend, Texas 65902


BIG BEND TELEPHONE COMPANY  
PO BOX 985  
BIG BEND, TX 65932-0985

*Service  
Summary*

June 1990

Long Distance Charges (North Star)	\$ 142.85
Local Telephone Charges	\$ 44.00
Equipment Charges	\$ 34.50
Federal, State and Local Taxes	\$ 28.76
Please Pay this Amount ==>	
	\$ 242.11

IF YOU HAVE QUESTIONS, PLEASE CALL 852-3381  
PLEASE MAKE CHECKS PAYABLE TO BIG BEND TELEPHONE COMPANY



**INTERSTATE TELEPHONE**  
LEADING THE WAY IN LONG DISTANCE CONNECTIONS

Billy Bob Smith  
RFD 23  
Big Bend, Texas 65902

Date	Time	City	Area Code and Number	Total Minutes	Charges
06/01/90	11:17 AM	Dallas, TX	(214) 555-6330	7.0	\$ 1.20
06/01/90	4:00 PM	Portland, OR	(503) 555-7435	65.1	\$ 11.30
06/01/90	6:39 AM	Greenville, MS	(601) 555-9534	25.6	\$ 7.85
06/03/90	11:30 AM	Podunk, UT	(801) 555-2630	13.5	\$ 7.65
06/05/90	8:50 PM	Cucamonga, CA	(213) 555-0534	14.6	\$ 9.25
06/06/90	10:20 PM	Buffalo, NY	(716) 555-7434	34.2	\$ 16.15
06/09/90	8:10 AM	Cripple Creek, CO	(303) 555-0332	11.0	\$ 4.20
06/09/90	9:01 AM	Natchez, MS	(601) 555-2134	2.7	\$ .75
06/09/90	12:14 PM	Podunk, UT	(801) 555-1737	5.5	\$ 2.90
06/11/90	4:34 PM	Long Beach, CA	(213) 555-7284	9.3	\$ 4.25
06/14/90	7:46 AM	Tombstone, AZ	(602) 555-1237	1.8	\$ 1.35
06/16/90	9:25 AM	Ouray, CO	(303) 555-8113	4.3	\$ 2.35

Figure 9. Example of a Customer's Phone Bill

## Input File

In this example, an application generates a spooled file containing statement data named PHONE. The application uses a printer file that specifies DEVTYPE(\*LINE). CRTAFPDTA is used to process the spooled file. Figure 10 on page 29 shows the data read by CRTAFPDTA.

```

Carriage
Control
|
Line 0 1 *...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8....+....9
0 1 2Billy Bob Smith RFD 23 Big Bend, Texas 65902 (214) 555-1234
![]L°Q[]01PHTOTL[]j[]
June 1990
Long Distance Charges (North Star) $ 142.85
5 Local Telephone Charges $ 44.00
Equipment Charges $ 34.50
Federal, State and Local Taxes $ 28.76

10 Please Pay this Amount ====> $ 242.11
1 3Billy Bob Smith RFD 23 Big Bend, Texas 65902 (214) 555-1234
![]L°Q[]01PHISTC[]j[]
![]L°Q[]01PHLDIS[]j[]
15 06/01/90 11:17 AM Dallas,TX (214) 555-6330 7.0 $ 1.20
06/01/90 4:00 PM Portland,OR (503) 555-7435 65.1 $ 11.30
06/01/90 6:39 AM Greenville,MS (601) 555-9534 25.6 $ 7.85
06/03/90 11:30 AM Podunk,UT (801) 555-2630 13.5 $ 7.65
06/05/90 8:50 PM Cucamonga,CA (213) 555-0534 14.6 $ 9.25
20 06/06/90 10:20 PM Buffalo,NY (716) 555-7434 34.2 $ 16.15
06/09/90 8:10 AM Cripple Creek,CO (303) 555-0332 11.0 $ 4.20
06/09/90 9:01 AM Natchez,MS (601) 555-2134 2.7 $ .75
06/09/90 12:14 PM Podunk,UT (801) 555-1737 5.5 $ 2.90
06/11/90 4:34 PM Long Beach,CA (213) 555-7284 9.3 $ 4.25
25 06/14/90 7:46 AM Tombstone,AZ (602) 555-1237 1.8 $ 1.35
06/16/90 9:25 AM Ouray,CO (303) 555-8113 4.3 $ 2.35
1 3Billy Bob Smith RFD 23 Big Bend, Texas 65902 (214) 555-1234
![]L°Q[]01PHISTC[]j[]
30 ![]L°Q[]01PHLDIS[]j[]
06/17/90 11:17 AM Mesquite,TX (214) 555-9707 4.0 $ 1.05
06/17/90 7:00 PM Crater Lake,OR (503) 555-9079 65.1 $ 11.30
06/19/90 8:39 AM Natchez,MS (601) 555-5204 34.4 $ 11.85
06/20/90 11:30 AM Podunk,UT (801) 555-4238 13.5 $ 7.65
35 06/21/90 8:50 PM Beverly Hills,CA (213) 555-8290 14.6 $ 9.25
06/22/90 8:10 AM Denver,CO (303) 555-1264 11.0 $ 4.20
06/22/90 10:20 PM Buffalo,NY (716) 555-8203 34.2 $ 16.15
06/23/90 8:01 AM Natchez,MS (601) 555-9234 2.7 $ .75
06/23/90 4:34 PM Cucamonga,CA (213) 555-3426 9.3 $ 4.25
40 06/23/90 12:14 PM Salt Flats,UT (801) 555-7228 5.5 $ 2.90
06/24/90 7:46 AM Tucson,AZ (602) 555-8233 1.8 $ 1.35
06/24/90 9:25 AM Punkin Center,CO (303) 555-5241 4.3 $ 2.95

```

Figure 10. Example Telephone Bill Input File

**Note:** There are some entries in the figure, such as

[]L°Q[]01PHTOTL[]j[], that have unprintable characters.

These are AFPDS records that reference overlays and are not displayable.

In Figure 10, the spooled file contains telephone statements that are dated May and June 1990. Each customer's statement has the same general format, although statements vary in size and number of pages. Assume you want the customer service representatives to be able to find a customer statement based on the customer's name, area code, phone number, or date. These values will be our index tags.

Because the statements are different lengths, CRTAFPDTA needs to be able to determine when a new statement begins. Each new statement has the customer's name on every page. The first page of every statement has the phrase "Local Telephone Charges". We will use the name and Local Telephone Charges as triggers.

## CRTAFPDTA Command

This is the command you use to index the statements:

```
CRTAFPDTA
FROMSPLF(PHONE)
TOSTMF('/MYOUTPUT/PHONE.AFP')
SPLNBR(*LAST)
FORMDF(*LIBL/F1BILL)
PAGDFN(*LIBL/P1BILL)
TOIDXSTMF('/MYOUTPUT/PHONE.IDX')
TORSCSTMF('/MYOUTPUT/PHONE.RSC')
IDXTRG((* 1 '1') (5 3 'Local Telephone Charges'))
IDXTAGFLD((1 2 20) (1 76 3) (1 81 8) (3 3 12) ('Telephone Bill'))
IDXTAG(('NAME' (*IDXTAGFLD1)) ('AREA' (*IDXTAGFLD2))
('PHONE' (*IDXTAGFLD3)) ('DATE' (*IDXTAGFLD4)))
IDXCDEPAG(500)
RSCDTA(*FORMDF *PAGSEG *OVL)
TOMRGSTMF('/MYOUTPUT/PHONE.MRG' *DLTINPUT)
```

Figure 11. Example CRTAFPDTA Command

Figure 11 specifies this information:

### FROMSPLF(PHONE)

PHONE is the input spooled file name.

### TOSTMF('/MYOUTPUT/PHONE.AFP')

Store the transformed data in the directory MYOUTPUT and name it PHONE.AFP.

### SPLNBR(\*LAST)

There are multiple spooled files named PHONE that this job created. Use the spooled file named PHONE with the highest number.

### FORMDF(\*LIBL/F1BILL)

Check the library list for a form definition named F1BILL. Use that form definition to specify information about the physical form.

### PAGDFN(\*LIBL/P1BILL)

Check the library list for a page definition named P1BILL. Use that page definition to arrange the data.

### TOIDXSTMF('/MYOUTPUT/PHONE.IDX')

Create a file that contains the indexing information. Call the file PHONE.IDX and store it in the directory MYOUTPUT.

### TORSCSTMF('/MYOUTPUT/PHONE.RSC')

Store all of the resources needed to view and print the file in a stream file called PHONE.RSC in the directory MYOUTPUT. The types of resources to store in this file are specified in **Resource Data** (RSCDTA).

### IDXTRG((\* 1 '1') (5 3 'Local Telephone Charges'))

These values together define the beginning of a new statement. (\* 1 '1') tells CRTAFPDTA to look in every record until it finds a 1 in column one. This is the indexing anchor record because it is specified first. (5 3 'Local Telephone Charges') tells CRTAFPDTA to look in the third column of the

fifth record from the indexing anchor record for the phrase "Local Telephone Charges". If both of these are found, this is the first page of a new statement. The new statement includes all of the following pages until the next trigger is found.

#### **IDXTAGFLD((1 2 20) (1 76 3) (1 81 8) (3 3 12) ('Telephone Bill'))**

These are the fields that are indexed. These fields contain the customer's name, area code, phone number, and bill date, respectively. They are all specified based on the indexing anchor record, where the indexing anchor record is considered record number one. For example, (1 2 20) tells CRTAFPDTA that the first index tag field is in the second column of the first record after the indexing anchor record and is 20 characters long.

The last entry, ('Telephone Bill'), is an example of using a literal value to index. Instead of pointing to a field to use, it adds the phrase "Telephone Bill" to the index on each phone bill.

#### **IDXTAG(('NAME' (\*IDXTAGFLD1)) ('AREA' (\*IDXTAGFLD2)) ('PHONE' (\*IDXTAGFLD3)) ('DATE' (\*IDXTAGFLD4)))**

This gives meaningful names to the tag fields defined in **Index field definition** (IDXTAGFLD). The first indexing tag specifies the customer's name, so it is labeled NAME, and so on. Notice that NAME and PHONE are the most unique indexing tags. You would use one of these values if you wanted to use the **Select index for group names** (IDXGRP) parameter. IDXGRP uses NAME by default because it is the first index tag.

#### **IDXCDEPAG(500)**

Use code page 500 for the index values and attribute names produced by CRTAFPDTA.

#### **RSCDTA(\*FORMDF \*PAGSEG \*OVL)**

Copy the form definition, the page segments, and the overlays that are needed to process this file into the resource stream file.

#### **TOMRGSTMF('/MYOUTPUT/PHONE.MRG' \*DLTINPUT)**

Merge PHONE.AFP, PHONE.RSC, and PHONE.IDX together. Name the new file PHONE.MRG and store it in the directory MYOUTPUT, then delete PHONE.AFP, PHONE.RSC, and PHONE.IDX.

Default values are used for the parameters that are not listed above.

### **Viewing the Indexed Output**

After you run the command shown in Figure 11 on page 30, you can open the file with AFP Viewer. In AFP Viewer, you can select a statement based on its values for one or more of these groups:

- NAME
- AREA (area code)
- PHONE
- DATE (month bill was issued)

For example, if you specify June for DATE and 214 for AREA, you get a list of all of the June statements for customers in area code 214. Double-click one of those statements to view it. For more information about using AFP Viewer, refer to *AS/400 Guide to Advanced Function Presentation and Print Services Facility*.

---

## E-mailing the CRTAFPDTA Output

This section describes how to use CRTAFPDTA with the PDF subsystem to split and e-mail indexed output. The PDF subsystem can take a spooled file, transform it into PDF files (one for each group), and e-mail each file to the appropriate receiver. If you want to group a spooled file with its resources and e-mail it without indexing the file first, you do not need to use CRTAFPDTA. Follow the instructions in “Example” on page 97.

### How it Works

1. CRTAFPDTA creates a Begin Named Group (BNG) structured field for each group defined by the **Index trigger definition** (IDXTRG) values. The BNG Groupname is the value specified through the **Select index for group names** (IDXGRP) and **Index tag definition** (IDXTAG) parameters. Groupname is used as the mail tag, so the group names must be either all e-mail addresses or all keywords. Use a PDF mapping program to map a keyword to an e-mail address.
2. When the file is sent to print on a properly configured device, the file is split at each BNG structured field. A PDF file is then created for each group.
3. If a PDF mapping program is specified on the PSF configuration object, the BNG Groupname value associated with each group should be a keyword. The keywords are interpreted by the PDF mapping program, which returns one or more valid e-mail addresses to which the file will be sent.
4. Each file is e-mailed using the protocol specified on the PSF configuration object.

### Steps to E-mail the Output

#### Before you begin:

Before you can successfully e-mail the indexed CRTAFPDTA output, you must have these:

- An appropriate PSF configuration object. The PSF configuration object must have these parameters specified:

**Generate PDF output**                      \*YES

**Multiple PDF files**                      \*YES \*SPLIT

**PDF user program (optional)**   *mapping-program-name*

**Library**                      *mapping-program-library*

**User resource library list (optional)**

\*JOBLIBL, \*CURLIB, or \*NONE

For information about creating the PSF configuration object, see “Setting up Your PSF Configuration Object” on page 95.

- A properly configured device. The device must be configured to use the PSF configuration object defined above. See “Configuring the Device” on page 96 for instructions.
- (optional) A PDF mapping program. You can create a PDF mapping program to map mail tags that are not valid e-mail addresses to valid e-mail addresses. For information about creating a PDF mapping program, see Appendix B, “PDF Mapping Program” on page 143 and Appendix C, “Templates” on page 155.

**To e-mail the CRTAFPDTA output, follow these steps:**



1. Use CRTAFPDTA to index the file. This inserts group tags in the file. See “Indexing with CRTAFPDTA” on page 26 for an example. Be sure that you specify these:
  - Use **Select index for group names** (IDXGRP) to specify the mail tags. You must ensure that these are either all e-mail addresses or all keywords. Keywords are mapped to e-mail addresses by a PDF mapping program. The example referenced above accepts the default value for **Select index for group names** (IDXGRP). The default value is the first **Index tag definition** (\*IDXTAG1), which is NAME in the example.
  - Use the **To stream file** (TOSTMF) parameter to specify that the output from the command is stored in QSYS.LIB as a physical file member.
  - Use **To index stream file** (TOIDXSTMF) to build the index. Specify TOIDXSTMF(\*DFT).
  - Use **Generate group names** (IDXGRPNAM) to specify that unique group names are not generated. Specify IDXGRPNAM(\*NO). If you do not specify this value, an 8-character numeric string is appended to the group name.
2. Use PRTAFPDTA to spool the AFP stream file (TOSTMF), then send it to print. The device you send it to must be set up to transform and e-mail IPDS files. See “Configuring the Device” on page 96 for instructions.

## Example

This example shows a file being split and e-mailed with CRTAFPDTA and the PDF subsystem. The file and indexing parameters from “Indexing with CRTAFPDTA” on page 26 are used. It uses a PDF mapping program that maps customer names to e-mail addresses. Specify the PDF mapping program in the PSF configuration object on the PDFMAPPGM parameter.

1. Index the file. This inserts the appropriate group tags. The value for NAME is used as the group name by default because it is the first indexing tag. The group name is specified by the **Select index for group names** (IDXGRP) parameter. The group name (in this case, NAME) is used as the mail tag. Because the mail tag is not a valid e-mail address, we need to use a PDF mapping program to map the names to the appropriate e-mail addresses. The user writes this program. MYLIB/MYOUTPUT must be a physical data file and must exist prior to running the CRTAFPDTA command. The member does not need to exist. If it does, it will be replaced by the new output.

```

| CRTAFPDTA
| FROMSPLF(PHONE)
| TOSTMF('QSYS.LIB/MYLIB.LIB/MYOUTPUT.FILE/PHONEAFP.MBR')
| SPLNBR(*LAST)
| FORMDF(*SPLF)
| PAGDFN(*SPLF)
| TOIDXSTMF(*DFT)
| IDXTRG((* 1 '1') (5 3 'Local Telephone Charges'))
| IDXTAGFLD((1 2 20) (1 76 3) (1 81 8) (3 3 12) ('Telephone Bill'))
| IDXTAG(('NAME' (*IDXTAGFLD1)) ('AREA' (*IDXTAGFLD2))
| ('PHONE' (*IDXTAGFLD3)) ('DATE' (*IDXTAGFLD4)))
| IDXGRPNAM(*NO)
|

```

2. Send the output AFP file to print on an appropriately configured device. See “Configuring the Device” on page 96 for instructions about setting up the device. In this example, PRINTER is an appropriately configured device and the form definition is the same one specified in the file PHONE’s spooled file attributes:

### Example

```
PRTAFPDTA FILE(MYLIB/MYOUTPUT) MBR(PHONEAFP) DEV(PRINTER) FORMDF(*LIBL/F1BILL)
```

## E-mail Considerations

When you e-mail the output from CRTAFPDTA, you should note this:

- The spooled file attributes for the original spooled file are not associated with the output files from CRTAFPDTA.
- When the output from CRTAFPDTA is printed, PSF looks for the resources in the library list, unless otherwise specified on the PSF configuration object **User resource library list** (USRRSCLIBL) parameter.

---

## CRTAFPDTA and ACIF

CRTAFPDTA implements functionality available to other platforms with ACIF. You can find more information about using the functions available through the CRTAFPDTA command in *AFP Conversion and Indexing Facility: User's Guide*, S544-5285.

This section contains two tables that list the CRTAFPDTA and ACIF parameters. Table 1 lists the CRTAFPDTA parameter first then lists the equivalent ACIF parameter (if one exists). Table 2 on page 35 lists the ACIF parameter first then lists the equivalent CRTAFPDTA parameter (if one exists).

*Table 1. CRTAFPDTA Parameters and Equivalent ACIF Parameters*

CRTAFPDTA Parameter	ACIF Parameter
AFPCHARS	CHARS
CTLCHAR - Cannot be specified directly to CRTAFPDTA. The value is obtained from the spooled file attribute CTLCHAR.	CC and CCTYPE
FORMDF	FORMDEF and FDEFLIB
FROMSPLF, JOB, SPLNBR, JOBSYSNAME, CRTDATE	INPUTDD
IDXCDEPAG	CPGID
IDXGRP	GROUPNAME
IDXGRPNAME	UNIQUEBNGS
IDXOBJ	INDEXOBJ
IDXPAGLMT	INDEXSTARTBY
IDXPAGNAM	DCFPAGENAMES
IDXTAG	INDEXn
IDXTAGFLD	FIELDn
IDXTRG	TRIGGERn
IGCSOSI - Cannot be specified directly to CRTAFPDTA. The value is obtained from the spooled file attribute IGCSOSI. <ul style="list-style-type: none"><li>• IGCSOSI(*YES) = PRMODE(SOSI1)</li><li>• IGCSOSI(*NO) = PRMODE(SOSI2)</li><li>• IGCSOSI(*RIGHT) = PRMODE(SOSI3)</li></ul>	PRMODE
IMAGEOUT	IMAGEOUT
PAGDFN	PAGEDEF and PDEFLIB
RSCDTA	RESTYPE

Table 1. CRTAFPDTA Parameters and Equivalent ACIF Parameters (continued)

CRTAFPDTA Parameter	ACIF Parameter
TBLREFCHAR - Cannot be specified directly to CRTAFPDTA. The value is obtained from the spooled file attribute TBLREFCHAR.	TRC
TOIDXSTMF	INDEXDD
TOMRGSTMF	N/A
TOSTMF	OUTPUTDD
TORSCSMTF	RESOBJDD

Table 2. ACIF Parameters and Equivalent CRTAFPDTA Parameters

ACIF Parameter	CRTAFPDTA Parameter
CC	Cannot be specified directly to CRTAFPDTA. The value is obtained from the spooled file attribute CTLCHAR.
CCTYPE	Cannot be specified directly to CRTAFPDTA. The value is obtained from the spooled file attribute CTLCHAR.
CHARS	AFPCHARS
COMSETUP	Not supported
CPGID	IDXCDEPAG
DCFPAGENAMES	IDXPAGNAM
FDEFLIB	Library portion of FORMDF
FIELDn	IDXTAGFLD
FONTECH	This parameter is not supported. Irrelevant for iSeries.
FONTLIB	This parameter is not supported. The requestor's library list is used for resource searches. The library list is augmented by the font libraries, if necessary.
FORMDEF	FORMDF
GROUPNAME	IDXGRP
IMAGEOUT	IMAGEOUT
INDEXDD	TOIDXSTMF
INDEXn	IDXTAG
INPUTDD	FROMSPLF, JOB, SPLNBR, JOBSYSNAME, CRTDATE
INDEXOBJ	IDXOBJ
INDEXSTARTBY	IDXPAGLMT
INDEXEXIT	Not supported
INPEXIT	Not supported
INPUTDD	FROMSPLF, JOB, SPLNBR, JOBSYSNAME, CRTDATE
MCF2REF	This parameter is not supported. CRTAFPDTA uses a value of CF for this parameter.

Table 2. ACIF Parameters and Equivalent CRTAFPDTA Parameters (continued)

ACIF Parameter	CRTAFPDTA Parameter
OBJCONLIB	Not supported
OUTEXIT	Not supported
OUTPUTDD	TOSTMF
OVLYLIB	This parameter is not supported. The requestor's library list is used for resource searches. The library list is augmented by the font libraries, if necessary.
PAGEDEF and	PAGDFN
PDEFLIB	Library portion of PAGDFN
PRMODE	Cannot be specified directly to CRTAFPDTA. The value is obtained from the spooled file attribute IGCSOSI. <ul style="list-style-type: none"> <li>• IGCSOSI(*YES) = PRMODE(SOSI1)</li> <li>• IGCSOSI(*NO) = PRMODE(SOSI2)</li> <li>• IGCSOSI(*RIGHT) = PRMODE(SOSI3)</li> </ul>
PSEGLIB	This parameter is not supported. The requestor's library list is used for resource searches. The library list is augmented by the font libraries, if necessary.
RESEXIT	Not supported
RESFILE	Not supported
RESOBJDD	TORSCSMTF
RESTYPE	RSCDTA
TRACEDD	Not supported. OS/400 provides the facility to perform a job trace. A job trace is equivalent to the trace provided by ACIF.
TRC	Cannot be specified directly to CRTAFPDTA. The value is obtained from the spooled file attribute TBLREFCHAR.
TRIGGERn	IDXTRG
UNIQUEBNGS	IDXGRPNAME
USERLIB	Not supported

---

## Chapter 3. Using the GIF to AFP Transform

This chapter describes how to install and use the **gif2afp** command. This command should be run from a DOS window. In this chapter, *command line* refers to the DOS command line. The chapter contains these sections:

- “Syntax”
- “Overview of gif2afp”
- “Installing the Image Transforms” on page 39
- “Planning for gif2afp” on page 39
- “Parameters” on page 41
- “Transforming GIF Data to AFP” on page 51
- “Error Recovery” on page 53
- “Limitations” on page 54
- “Acknowledgments” on page 54

---

### Syntax

```
gif2afp [-a ImageType] [-alg ProcessingAlgorithms]  
[-calib calibration] [-C ConfigurationFile]  
[-clean cleanup] [-cmp compression]  
[-crop CropFactors] [-fit {trim | scale}]  
[-gcorr GrayscaleMappingTable] [-ink color] [-inv]  
[-j ScanOffsetFileName] [-l ImageLength ]  
[-M MemoryBound] [-ms space] [-msf SpaceFraction]  
[-o OutputFile] [-outbits NumberOfOutputBits]  
[-outcolor OutputColorModel] [-p PageRange]  
[-pagetype PageType] [-paper PaperSize]  
[-r resolution] [-respath ResourceSearchPath]  
[-rot rotation] [-scale ImageSize]  
[-sgcorr ScannerCorrection]  
[-sniff | -nosniff] [-term | -noterm]  
[-thresh HalftoneFile] [-v | -nov] [-w ImageWidth]  
[-x LeftMargin] [-y TopMargin] [-z]  
[file]
```

Figure 12. gif2afp command syntax

---

### Overview of gif2afp

The GIF to AFP transform is a command controlled transform that runs on a PC. It converts graphics interchange format (GIF) data to an AFP overlay, page segment, image object, or document, or to a Postscript Level 2 data stream file. Use the pagetype parameter to specify the type of output you want. You can use either of these tools to turn the output AFP resource (overlay or page segment) into a resource on the OS/400:

- iSeries Access. Refer to *AS/400 Guide to Advanced Function Presentation and Print Services Facility* for instructions.
- OS/400 commands. See “Turning a PC AFP Resource into an OS/400 Resource” on page 141.

The transform processes bilevel (black and white), grayscale and color GIF images. GIF files can contain multiple images. The output is bilevel, 4-bit or 8-bit grayscale, or 24-bit YCbCr color. The transform automatically uses halftoning to convert the grayscale and color images to bilevel.

This transform runs on Windows NT<sup>®</sup> service pack 4, Windows 2000, Windows 98 and Windows 95.

## Output Data Types

This section describes valid output data types.

The output data stream is MO:DCA IS/1, MO:DCA IS/2, or PostScript Level 2. Use the **-a** option to specify the output data stream type. IS/1 images are bilevel and are encoded as IOCA Function Set 10 (FS10) or IM1. IM1 images are uncompressed. IOCA FS10 output images can be uncompressed or compressed by one of the four available compression algorithms (see the **-cmp** option).

IS/2 images can be either bilevel, 4-bit or 8-bit grayscale, or 24 bit YCbCr color. IS/2 images are encoded as IOCA FS11, FS42, or FS45. Multi-bit images are encoded as IOCA FS11 or FS45.

PostScript Level 2 images can be uncompressed or ITU-T T.6 Group 4 bilevel, uncompressed 8-bit grayscale or uncompressed 24-bit RGB color. By default, the transform leaves scaling and halftoning to the printer, that is, color GIF images are output as 24-bit RGB color images.

If the input image is grayscale or color and the output image type is bilevel, the transform internally scales the image to the desired size first. It then uses a halftoning algorithm to convert it to bilevel.

## Scaling

To maintain image quality, scaling is done before halftoning. The options **-alg**, **-clean**, and **-ink** control scaling algorithms. The size and position of output image depend on several factors:

- The size specified to the transform
- The form map specified at print time
- Forms and options loaded in the printer

If the image is too large for page, it is cropped by default. You can override this with the **-fit scale** option. If paper size is not specified, the default paper size is increased to fit the image.

To explicitly specify the image size, use **-scale**. **-scale orig** forces the output image to be the same size as the original image in most cases. An image has to be scaled if the original image is being halftoned or has different resolutions in the X and Y directions (non-square pixels). By default, the transform leaves scaling to the printer. There are two exceptions:

- Images that are being halftoned, which are always scaled in the transform
- Images that have non-square pixels

**Note:** Because GIF files do not specify resolutions, the image is assumed to have the resolution of the output device.

## Halftoning

When necessary, the transform uses a halftoning algorithm to convert the image to bilevel. When an image is converted to bilevel, the characteristics of the output device such as dot shape and dot gain must be taken into consideration. The internal grayscale mapping table is optimized for the 600-pel IBM Infoprint 4000 laser printer. The **-gcorr** option can be used to specify a different mapping of gray levels.

The **-alg** option controls halftoning algorithms. For very light or dark images, halftoning algorithms can be automatically re-calibrated. Halftoning algorithms are very computation-intensive and slow. You will notice a decrease in performance when images are halftoned.

---

## Installing the Image Transforms

In order to use the GIF to AFP transform, you must install it on your PC. To install the image transforms, follow these steps:

1. Right-click Network Neighborhood and select **Find Computer...** Specify the name of your OS/400 in the **Named:** field.
2. When Windows finds your OS/400, double-click your system to display the contents.
3. Expand QIBM → ProdData → InfoprintServer → Transforms → Install → Image.
4. Double-click **setup.exe** and follow the directions.

**Note:** Installing the image transforms allows you to use the GIF to AFP, JPEG to AFP, and TIFF to AFP transforms.

---

## Planning for gif2afp

Before using the transform, do the following:

- Set up a configuration file - optional. See “Setting Up a Configuration File”.
- Set up environment variables - optional. See “Using Environment Variables” on page 40.
- Edit your Windows path - optional. See “Editing Your Windows Path” on page 41.
- Ensure that you have the proper authorities. See “Authorities” on page 41.

## Setting Up a Configuration File

The configuration file is one place you can specify options for the transform. Entries are of the form *option=value*, where each pair is on a separate line. Unless otherwise noted, options and other syntax are the same for entries in the configuration file as for the command line arguments. You cannot specify input files in the configuration file.

The following options cannot be used in the configuration file:

- **-calib**
- **-C**

Options without values, such as **-v**, must be given the value **yes**.

Specify the configuration file on the command line with the **-C** option. If you do not specify one, the transform checks for the default configuration file **C:\Program Files\IBM Printing Systems\gif2afp\gif2afp.cfg** and uses that, if it exists.

The following is an example of a valid configuration file. It specifies that the output image is 8.5 by 11 inches and the resolution of the output device is 300 dpi. The output filename is myfile.afp and it is stored in C:\mydirectory. Verbose mode is on, and the output is a page segment. The spaces surrounding the equals ( = ) sign are optional, but if one is specified after the equals sign on a parameter that specifies a file name, the space is included in the file name.

```
scale = letter
r = 300
o =C:\mydirectory\myfile.afp
v = yes
pagetype = pseg
```

**Note:** Parameters specified in the configuration file are overridden by parameters specified on the command line.

## Using Environment Variables

You can use environment variables to specify parameters for the GIF to AFP transform. Environment variables have the form `Variable: GIF2AFP_option`, `Value: value`, where *option* and *value* are the same as the command line and configuration file option and value. If the option does not have a value, specify a blank surrounded by quotation marks ( " ").

You cannot use **-calib** as an environment variable.

To achieve the same effect as the configuration file in the last example, add the following statements to your environment variables on your Windows system:

VARIABLE	VALUE
GIF2AFP_scale	letter
GIF2AFP_r	300
GIF2AFP_o	C:\mydirectory\myfile.afp
GIF2AFP_v	" "

To edit your Windows environment variables, except with Windows 98, follow these steps:

1. Right-click **My Computer** and select **Properties**.
2. Select the **Environment** page.
3. Specify the parameters you want by repeating these steps as many times as necessary:
  - a. Fill in **Variable** and **Value** for the parameter you want to use.
  - b. Click **Set**.
4. Click **Apply**.
5. Restart your computer for the changes to take effect.

To edit your Windows environment variables for the current session with Windows 98, follow these steps:

1. Open a DOS prompt.
2. Specify the parameters you want by entering this command as many times as necessary: `set variable=value`. For example to set GIF2AFP\_scale to letter, specify `set GIF2AFP_scale=letter`.
3. Exit the DOS command prompt.

To edit your Windows environment variables for every session, put the lines described in step 2 in your Autoexec.bat file.



**Note:** Environment variables are overridden by parameters specified in the configuration file or on the command line.

## Editing Your Windows Path

If you put the directory that contains the transform executable in your Windows path, you can run the transform command from any directory. If you installed the transforms in the default directory and you run the transforms from the C directory, you do not have to edit the path. Follow these steps to edit the path, except with Windows 98.

1. Right-click **My Computer** and select **Properties**.
2. Select the **Environment** page.
3. In the System Variables area, click on the current path.
4. Add the new path to the end in the format `VALUE1;VALUE2;`. For example, if you installed the transforms in the default directory, add this to the end of the current path: `c:\Program Files\IBM Printing Systems\bin;`.
5. Click **Apply**.
6. The changes take effect immediately.

To edit your path with Windows 98, follow these steps:

1. Open `C:\Autoexec.bat` in an editor.
2. Add this line to the end of the file: `SET PATH=directory;`. For example, if you installed the transforms in the default directory, add this to the end of the file: `SET PATH=c:\Program Files\IBM Printing Systems\bin;`.

## Authorities

You need these authorities to use this transform:

- Input file – read authority
- Output file – write authority

---

## Parameters

This section describes how to use the **gif2afp** parameters.

**-a {ioca | ioca10 | ioca11 | ioca42 | ioca45 | im1 | PS.2}**

Specifies the output image type. See also **-alg**.

**ioca** The output image type is IOCA FS10.

Make sure that your printer supports IOCA before choosing this value. If your printer does not support IOCA, choose **im1**.

**ioca10**  
The output image type is IOCA FS10.

Make sure that your printer supports IOCA before choosing this value. If your printer does not support IOCA, choose **im1**.

**ioca11**  
The output image type is IOCA FS11. IOCA FS11 cannot be used with an IPDS printer.

**ioca42**  
The output image type is IOCA FS42.

Make sure that your printer supports IOCA before choosing this value. If your printer does not support IOCA, choose **im1**.

## **ioca45**

The output image type is IOCA FS45.

Make sure that your printer supports IOCA before choosing this value. If your printer does not support IOCA, choose **im1**. If you use this value, JPEG is the default output compression type and the image is subsampled to 300 dpi. If **-cmp lzw** is specified, tLZW is used to compress the image and it is not subsampled.

## **im1**

IM1 images are uncompressed and therefore use more space. The processing might also take longer because IM1 images cannot be scaled by the printer and internal scaling algorithms are used.

## **PS.2**

The output image type is PostScript Level 2.

## **-alg {afp | alg1 | alg2 | alg3}, {htod1 | htfs}, [htcal1]**

Specifies scaling and halftoning algorithms. You can specify multiple values, separated by commas, for one **-alg** option. The effect is the same as specifying multiple **-alg** options. For example, **-alg htfs,htcal1** is equivalent to **-alg htfs -alg htcal1**.

The following values specify scaling algorithms:

- afp** Scaling is done by the default algorithm in the printer. This makes the transform faster. If the image must be reduced, however, this algorithm might lose some image information, such as thin lines. This value is the default for bilevel images with IOCA output.
- alg1** Scaling is done by the internal scaling algorithm. This algorithm will not discard any ink in the image. White space in the image, however, might be lost. This algorithm makes the transform run longer.
- alg2** Scaling is done by a more flexible version of **alg1**. If this value is selected, **-clean** can be used to specify the amount of ink cleanup that is performed.
- alg3** Scaling is done by a scaling algorithm that works by deleting or duplicating rows and columns in the image. This is a fast general-purpose algorithm. This value is the default for bilevel images with IM1 output.

The following values specify halftoning algorithms. Each halftoning algorithm uses a different internal default calibration curve. If automatic calibration is turned on, a different set of calibration curves is used.

- htod1** Halftoning is done using an ordered dither with a screen derived from the value of the **-thresh** option. The default is an 85 line per inch screen.
- htfs** Halftoning is done by the Floyd-Steinberg algorithm.
- htcal1** Re-calibrates the halftoning algorithm for each image. **htcal1** forces the transform to read the whole image into memory and requires an additional pass through the image. It should be used only for very light or very dark images.

See also **-clean**, **-gcorr**, **-ink**, **-paper**, **-scale**, **-thresh**, **-l**, **-w**, **-x**, and **-y**.

## **-calib {scanner | printer | patch | patchr}, name1...**

Specifies the transform calibration for the scanner or printer. You cannot specify **-calib** in the configuration file or as an environment variable.

<b>scanner</b>	The file is calibrated for the scanner. If this value is specified, you must also specify <b>-sgcorr</b> .
<b>printer</b>	The file is calibrated for the printer.
<b>patch</b>	If this value is specified, any value specified for <b>-sgcorr</b> is ignored.
<b>patchr</b>	If this value is specified, any value specified for <b>-sgcorr</b> is ignored.
<i>name1</i>	Specifies the name of the file being calibrated. You can specify multiple files.

#### **-C** *ConfigurationFileName*

Specifies the name of a configuration file. If multiple **-C** options are given on the command line, they are processed in the order in which you specified them. The default configuration file is **C:\Program Files\IBM Printing Systems\gif2afp\gif2afp.cfg**.

#### *ConfigurationFileName*

The name and location of a configuration file. If there are spaces in the path or name, enclose the string in double quotation marks. For example, "C:\Program Files\IBM Printing Systems\files\file.gif".

#### **-clean {0.5 | x.xxx}**

Specifies when to turn on a pel. The scaled pel is turned on if the specified part of its area, defined by the threshold, is painted in the original image. The threshold is the value **0.5** or *x.xxx*. It is a decimal that defines a fraction. If the fraction of a pel that is painted in the original image is greater than this threshold, the pel is turned on. For example, if you specify **-clean .4**, a pel is turned on any time more than 4 tenths, or 20%, of the pel is painted in the original image. This option is only used if **-alg alg2** is specified.

**0.5** The threshold is .5. **alg2** turns on a pel any time more than half of it is painted in the original image.

*x.xxx* The threshold is *x.xxx*. *x.xxx* is a number in the range of 0-1.

If **-clean 0.0** is specified, **alg2** turns on a pel any time any part of the pel is painted in the original image. Therefore, it behaves like **alg1**.

**-clean 1.0** turns on a pel when more than 100% of the pel is painted in the original image; therefore, it results in a blank image.

#### **-cmp {g4|none|lzw|mh|mmr|g3|jpeg|abic }**

Specifies the algorithm to use to compress the output image.

**g4** ITU-T T.6 Group 4 algorithm. This is the default compression for bilevel images. Make sure that your printer supports the Group 4 compression algorithm before specifying this value.

**lzw** Lempel-Zif-Welch algorithm.

**none** The output is uncompressed. This is the default for non-JPEG grayscale and color images.

**mh** ITU-T T.4 Group 3 Modified Huffman algorithm

**mmr** IBM Modified Modified Read algorithm

**g3** ITU-T T.4 Group 3 Modified Read algorithm

**jpeg** JPEG nondifferential Huffman coding with baseline DCT algorithm. This algorithm can only compress 8-bit grayscale and 24-bit color images.

**abic** ABIC algorithm

**-crop *t,b,l,r* [*d|i|m|p*]**

Specifies image cropping. Cropping is done in the context of the output page, so that “top” means the top of the output page, regardless of the **-rot** option. The resulting image has the size specified by the **-scale** option. If you want the image cropped from at least one direction, you must specify all of the commas. For example **-crop ,5,,4** and **-crop 0,5,0,4** tell the transform to crop the image five dots from the bottom and four dots from the right. **-crop 5,4** gives you an error. See also **-j**.

The following values specify the amount of cropping:

- |          |  |
|----------|--|
| <i>t</i> | Specifies how much should be cropped from the top edge. Designate a positive or negative decimal value. A negative number indicates that a blank space is inserted.    |
| <i>b</i> | Specifies how much should be cropped from the bottom edge. Designate a positive or negative decimal value. A negative number indicates that a blank space is inserted. |
| <i>l</i> | Specifies how much should be cropped from the left edge. Designate a positive or negative decimal value. A negative number indicates that a blank space is inserted.   |
| <i>r</i> | Specifies how much should be cropped from the right edge. Designate a positive or negative decimal value. A negative number indicates that a blank space is inserted.  |

The following values specify the unit of measurement:

- |          |                                  |
|----------|----------------------------------|
| <b>d</b> | Measurements are in dots (pels). |
| <b>i</b> | Measurements are in inches.      |
| <b>m</b> | Measurements are in millimeters. |
| <b>p</b> | Measurements are in points.      |

**-fit {trim | scale}**

Specifies what the printer does when the image is too large to fit on the paper. Specify paper size with the **-paper** option.

**trim** The image is positioned and then trimmed to make it smaller.

**scale** The image is scaled to make it smaller.

**-gcorr *FileName***

Specifies a file that contains a grayscale mapping table for halftoning grayscale and color images. The mapping table must be provided to compensate for the printing characteristics of the particular output device and paper type, such as dot gain. This file must contain one of these:

- 256 real numbers, one for each level of gray in the 8-bit grayscale image
- The PostScript **settransfer** operator

The algorithm converts every image to 8-bit grayscale before applying the halftoning algorithm. The default internal mapping table is optimized for the 600-pel IBM Infoprint 4000 laser printer. Only experienced users should change this. PostScript code submitted by the **-thresh** option can have the same effect as **-gcorr**. See also **-alg** and **-thresh**.

*FileName* Specifies the name and location of a file that contains a grayscale mapping table. If there are spaces in the file name

and you are specifying this value on the command line, enclose the file name in double quotation marks.

**-ink { black | white }**

Specifies whether the transform preserves black or white pels when compressing. The internal scaling algorithm needs to know, because “ink” is preserved at the expense of “non-ink” This option is used only with scaling algorithm **alg1** or **alg2**.

If the option **-inv** is used to print the reversed image, the inversion is done before the transform begins processing. Therefore, black or white must be specified in the terms of the printed image, not in the terms of the input GIF image.

**black** The image is black on white.

**white** The image is white on black.

**-inv**

Reverses the image. Areas that are black in the original image become white and vice versa. Note that the reversed image might be black on white or white on black, depending on the original image.

**-j *ScanOffsetFileName***

Specifies the location and name of the scan offset file. If there are spaces in the path and you are specifying this value on the command line, enclose the path in double quotation marks. Sometimes, input images must be shifted to get the desired positioning on the page. The scan offset file contains the directions for shifting such images. The file has this format:

```
[attribute]
values
.
.
.
values
[attribute]
values
.
.
.
values
```

These are the recognized attributes:

**[Units]**

Allowed values are: millimeters, **mm**, **inches**, **in**, **points**, **dots**, and **pels**

**[Page\_Offset\_Type]**

Allowed values are: **crop** and **grow**

**[Page\_Offsets]**

The values are in the format *FileName*, **H***horiz*, **V***vert* Each entry refers to an input file. The file names or the leading commas must be specified, but are currently ignored. The offset factors are applied to each file in turn. The same factors are applied to each image in the file. The *horiz* and *vert* values are the amount that the image is shifted in the horizontal and vertical directions. The positive directions are down and right, the origin is in the left upper corner of the page. The shifts are specified context of the output page.

Extra spaces and blank lines are allowed. Unrecognized attributes and their values are ignored.

If the image is moved left or up, part of the bitmap is removed. If the image is moved down or right, removal is governed by the value of the **[Page\_Offset\_Type]** attribute.

**-l {11i | *length* [d | m | i | p]}**

Specifies the paper length. See also **-w** and **-paper**.

11i The paper is 11 inches long.

*length* Specifies the length of the paper. Give the length in decimal format.

The following values specify the unit of measurement:

d The length is measured in dots. This value only uses the whole number part of a decimal value.

m The length is measured in millimeters.

i The length is measured in inches.

p The length is measured in points.

**-lut *FileName***

Specifies a color mapping look up table. The table must be appropriate for the file's color type. For example, if your file is RGB, the look up table must be for RGB mapping.

*FileName*

Specifies the name and location of a color mapping look up table. If there are spaces in the path and you are specifying this value on the command line, enclose the path in double quotation marks.

**-M {0 | *nnn*}**

Specifies the amount of memory available to **gif2afp**.

0 There is no memory limit.

*nnn* Specifies the maximum amount of memory allocated to **gif2afp**, in kilobytes. *nnn* is a value in the range of zero to the maximum value that Windows allows.

**-ms {10 | *nnn*}**

Specifies the amount of disk space that must be available on the file system that contains the output file. This limit is enforced every time data is written to the output file. If the output is stdout, this option is ignored. See also **-msf**.

10 There must be at least 10 kilobytes of disk space available on the file system that contains the output file.

*nnn* There must be at least *nnn* kilobytes of disk space available on the file system that contains the output file.

**-msf {0.1 | *x.xxx*}**

Specifies the minimum amount of available space that must be on the file system containing the output file. The space is given as a fraction of the total space in the file system. This limit is enforced every time data is written to the output file. If the output is stdout, this option is ignored. The default is **0.1**. See also **-ms**.

0.1 One-tenth of the total space in the file system must be available.

**x.xxx** Specifies a number in the range of 0-1 that designates the portion of the space that must be available.

**[-nosniff | -sniff]**

Specifies whether bilevel images are automatically converted to black on white. The conversion algorithm determines the number of black and white bits. If there are fewer white bits, it assumes that the image is white on black, and inverts the image to be black on white. This algorithm is run after the conversion specified by the Photometric Interpretation tag is applied (if any).

This option is ignored if the image is not bilevel.

**-nosniff** Bilevel images are not converted to black on white.

**-sniff** Bilevel images are converted to black on white.

**[-noterm | -term]**

Specifies whether error recovery is enabled. When error recovery is enabled, errors in GIF files are treated as nonfatal. The transform tries to recover with the next image in the file. If error recovery is disabled and the transform finds an error in a GIF file, the transform ends with an error message and a nonzero return code.

**-noterm** Enables error recovery.

**-term** Disables error recovery.

**[-nov | -v]**

Specifies whether verbose mode is on. When verbose mode is on, the transform sends a message as it opens each resource file. It then echoes the command line and sends a message for each file name as it is processed. When verbose mode is off, the transform sends only error messages.

**-nov** verbose mode is off.

**-v** verbose mode is on.

**-o FileName**

Specifies the output file name. This is optional. If you do not specify an output file name or request standard output, the AFP file is stored in the same location as the input file with the same name as the input file, but with .afp as the extension.

**FileName** The output file name and location. If there are spaces in the path and you are specifying this value on the command line, enclose the path in double quotation marks. Specify - or **stdout** for stdout. No file is created if you specify stdout.

If you want to use this file on your OS/400 system, specify `\\system\QDLS\file.afp`, where *system* is the name of your OS/400 system and *file* is the name of the output file.

For information about importing the AFP file from your PC to your OS/400 system, see "Turning a PC AFP Resource into an OS/400 Resource" on page 141 or refer to for instructions to use iSeries Access.

**-outbits {1 | 24 | 4 | 8}**

Specifies the number of bits per pel in the output.

**1** The output has 1 bit per pel. If the output image type is IOCA FS10 or IM1, the number of output bits is set to 1 and this option is ignored.

**24** The output has 24 bits per pel. This is the default for IOCA FS11

output. This number represents the maximum number of bits used per pel. Thus, even if the number of bits is set to 24, a 4-bit grayscale image is still displayed in 4-bit format.

**4** The output has 4 bits per pel.

**8** The output has 8 bits per pel.

**-outcolor {ycbcr | rgb | ycrcb}**

Specifies the output color model. This value is ignored unless the image type (specified with **-a**) is IOCA FS11 and the number of output bits (specified with **-outbits**) is 24.

**ycbcr** The output color model is YCbCr.

**rgb** The output color model is RGB.

**ycrcb** The output color model is YCrCb.

**-p {even | odd | nn | nn-mm | nn-}**

Specifies which pages the output should contain. If no **-p** option is given, all the pages are output. You can specify **-p** multiple times. Their effect is cumulative. The pages are always printed in ascending order and are numbered starting with one.

**even** Print all even pages.

**odd** Print all odd pages.

**nn** Print page *nn*.

**nn-mm**  
Print pages *nn* to *mm*.

**nn-** Print all pages starting with page *nn*.

**-pagetype {page | overlay | ovly | pseg | object}**

Specifies the output page type. The same page type is set for every page in the output. However, you can use **-p** to output only specified pages. See also **-a**.

**page** All output is pages.

**overlay** All output is overlays.

**ovly** All output is overlays.

**pseg** All output is page segments.

**object** All output is objects.

**-paper {letter | a5 | a4 | a3 | folio | legal | ledger | xxx.xx,yyy.yy[\_d | m | i | p]}**

Specifies the output paper size. If the paper size is not specified, the paper size is increased, if necessary, to contain the output image.

See also **-l**, **-w**, and **-scale**.

**letter** 8.5 by 11 inches (216 by 279 mm)

**a3** A3 format (297 by 420 mm or 11.69 by 16.54 inches)

**a4** A4 format (210 by 297 mm or 8.27 by 11.69 inches)

**a5** A5 format (148 by 210 mm or 5.83 by 8.27 inches)

**folio** 8.5 by 13.0 inches (216 by 330 mm)

**legal** 8.5 by 14.0 inches (216 by 356 mm)

**ledger** 11.0 by 17.0 inches (279 by 432 mm)



*xx.xx,yy.yy* [**d | i | m | p**]

Specifies the page dimensions.

*xx.xx* Specifies the horizontal page dimension.

*yy.yy* Specifies the vertical page dimension.

The following values specify the unit of measurement:

**d** Measurements are in dots (pels).

**i** Measurements are in inches.

**m** Measurements are in millimeters.

**p** Measurements are in points.

**-r {300 | *nnn*}**

Specifies the output device resolution in dots per inch.

**300** The output device resolution is 300 dots per inch.

*nnn* The output device resolution is *nnn* dots per inch.

**-respath** *directory*[:*directory...*]

Specifies the search path for resource files, such as the scan offset files and grayscale calibration curve files. If a resource file name is specified as a relative name, the transform searches every directory in the path, in the order specified in the path, until it finds the file or fails. This search path does not apply to anything in the configuration file.

*directory* Specifies the search path for resource files. The default is **C:\Program Files\IBM Printing Systems\config**.

**-rot {0 | **90** | **180** | **270** [**p** | **i**]}**

Specifies how far to rotate the image and where the rotation is done. The transform or the printer can rotate the image.

These values specify how far to rotate the image:

**0** The image is not rotated.

**90** The image is rotated clockwise 90 degrees.

**180** The image is rotated clockwise 180 degrees.

**270** The image is rotated clockwise 270 degrees.

These values specify where the image is rotated:

**p** The printer rotates the image.

**i** The transform rotates the image. When **i** is specified, the transform takes longer to run and uses more memory.

**-scale {orig | **a5** | **a4** | **a3** | **letter** | **folio** | **legal** | **ledger** | *xxx%* | *xxx.xx,yyy.yy* [**d** | **m** | **i** | **lp**]}**

Specifies the output image size. The image is scaled to this size and centered in the area defined by the paper size and margin options.

**orig** Preserve the input image size, regardless of the printable paper area. The image still has to be scaled if the input file specifies non-square pels or is halftoned.

**a3** A3 format (297 by 420 mm or 11.69 by 16.54 inches)

**a4** A4 format (210 by 297 mm or 8.27 by 11.69 inches)

**a5** A5 format (148 by 210 mm or 5.83 by 8.27 inches)

**letter** 8.5 by 11 inches (216 by 279 mm)

**folio** 8.5 by 13.0 inches (216 by 330 mm)

**legal** 8.5 by 14.0 inches (216 by 356 mm)

**ledger** 11.0 by 17.0 inches (279 by 432 mm)

**xxx%** Scaling is relative to the original image size. Numbers below 100% mean reduction, while numbers above 100% mean enlargement. For example, **-scale 200%** causes the output image to be twice as long and twice as wide as the input image.

**xx.xx,yy.yy[d | m | i | p]**

Specifies the image dimensions.

**xx.xx** Specifies the horizontal dimension.

**yy.yy** Specifies the vertical dimension.

The following values specify the unit of measurement:

**d** Measurements are in dots (pels).

**i** Measurements are in inches.

**m** Measurements are in millimeters.

**p** Measurements are in points.

**-sgcorr** *FileName*

Specifies the name of the scanner-related calibration file. If **-calib** is not specified, this option is ignored.

*FileName* The location and name of the scanner calibration file. If there are spaces in the path and you are specifying this value on the command line, enclose the path in double quotation marks.

**-sniff**

See **[-nosniff | -sniff]**

**-term**

See **[-noterm | -term]**

**-thresh** *FileName*

Specifies a file that contains a PostScript Type 1 or Type 3 halftone dictionary. Alternatively, the PostScript code can specify the **setscreen** operator instead of a Type 1 dictionary. The halftone cell in the dictionary is used to overwrite the default ordered dither clustered dot halftone cell. If the PostScript code contains a transfer function, either in the halftone dictionary or specified by the **settransfer** operator, the current grayscale correction curve is overwritten as if **-gcorr** was used.

*FileName* The name and location of the file that contains an appropriate halftone dictionary. If there are spaces in the path and you are specifying this value on the command line, enclose the path in double quotation marks.

**-v** See **[-nov | -v]**

**-w {8.5i | yyy.yy[d | m | i | p]}**

Specifies the paper width. See also **-l** to set the paper length or **-paper** to set both dimensions at the same time.

**8.5i** The page width is 8.5 inches.

The following values specify the unit of measurement:

**d** The width is measured in dots (pels).

**i** The width is measured in inches.

**m** The width is measured in millimeters.

**p** The width is measured in points.

**-x {0 | *nnn.nn*[**d** | **m** | **i** | **p**]}**

Specifies the left margin.

**0** There is no left margin.

*nnn.nn*

The left margin is *nnn.nn* units.

The following values specify the unit of measurement:

**d** The margin is measured in dots (pels).

**i** The margin is measured in inches.

**m** The margin is measured in millimeters.

**p** The margin is measured in points.

**-y {0 | *nnn.nn*[**d** | **m** | **i** | **p**]}**

Specifies the top margin.

**0** There is no top margin.

*nnn.nn*

The top margin is *nnn.nn* units.

The following values specify the unit of measurement:

**d** The margin is measured in dots (pels).

**i** The margin is measured in inches.

**m** The margin is measured in millimeters.

**p** The margin is measured in points.

---

## Transforming GIF Data to AFP

This section describes how to transform GIF data to AFP and gives examples.

### Input to the GIF to AFP Transform

Input can be a GIF file submitted by standard input (stdin) or specified on the command line. If no input file is specified, stdin is assumed. The extension .gif or .GIF does not need to be given explicitly. **gif2afp** first tries to open the file as specified and, if unsuccessful, tries to append the extensions .gif and .GIF in turn.

### Specifying the Output File

The output file name and location can be derived from the input file name and location, specified on **-o**. Alternatively, you can force output to be standard output (stdout - this is typically output to your screen) by specifying **-o -** or **-o stdout** on the command line or **o = -** or **o = stdout** in the configuration file. If you request

standard output, no output file is generated. By default, the output file has the same name as the input file and is stored in the same location as the input file.

If the output file name is not specified explicitly, the transform strips any extension from the input file name and appends the appropriate extension to create the output file name.

For example, all the following commands have `myfile.afp` as the output file:

```
gif2afp myfile
gif2afp myfile.gif
gif2afp myfile -o myfile.afp
```

Note that there is no requirement for the explicitly specified input and output files to have `.gif` and `.afp` extensions. To process GIF file `foo.bar` into an AFP file named `foo.bar2`, use this command to invoke the transform:

```
gif2afp foo.bar -o foo.bar2
```

## Invoking gif2afp

To invoke **gif2afp**, from a DOS prompt specify `gif2afp file -option value.... file` is the location and name of the GIF file and *option value* is any parameter you want to specify on the command line. You do not have to specify any parameters on the command line. The parameter values are determined by the following hierarchy (least significant first):

1. Internal defaults
2. Environment variables
3. Configuration file
4. Command-line arguments

### Notes:

1. If there are spaces in the path or name, enclose the string in double quotation marks. For example, "C:\Program Files\IBM Printing Systems\files\file.gif".
2. If the directory that contains **gif2afp.exe** is not in your PATH specification, you must be in that directory to run the command. If you accepted the default directories at install time, this directory is **C:\Program Files\IBM Printing Systems\bin**. See "Editing Your Windows Path" on page 41 for instructions to change your PATH.

## Command Line Arguments

When using the command line, note the following:

- Options and input file names can be specified in any order.
- Only file names are case-sensitive.
- If an option is specified multiple times, only the last one is recognized.
- If multiple configuration files are given, they are processed in the order in which you specify them.

## Examples

### Example 1

This example transforms a GIF file called `MyOverlay.gif` into an AFP overlay called `MyOverlay.afp`. The output file name is not specified because it is given the same name as the input file with the `.afp` extension by default. It is stored in the specified directory so it can easily be imported to the system `MyIseries`. Enter this command at a DOS prompt:

```
gif2afp MyOverlay.gif -pagetype overlay -o \\MyIseries\QDLS\
```

The same results can be achieved with this command and configuration file:

```
gif2afp MyGifFile.gif -C ConfigFile1
```

ConfigFile1 contains this:

```
pagetype = overlay  
o = \\MyIseries\QDLS\
```

## Example 2

This example transforms a GIF file called BigLogo.gif into an AFP page segment called LOGO. The logo needs to be scaled to 2 inches by 2 inches and rotated 90 degrees. To have greater control over which pels are colored, we specify that the alg2 algorithm is to be used to scale the image. We also specify that a pel in the output is only colored if 80% of the pel was colored in the input file. However, specifying alg2 makes the transform take longer to complete. The output is stored in the specified directory so it can easily be imported to the system MyIseries. Enter this command in a single line at a DOS prompt:

```
gif2afp BigLogo.gif -pagetype pseg -scale 2,2i -rot 90 -alg alg2 -clean .8  
-o \\MyIseries\QDLS\LOGO.afp
```

The same results can be achieved with this command and configuration file:

```
gif2afp BigLogo.gif -C ConfigFile2
```

ConfigFile2 contains this:

```
pagetype=pseg  
scale=2,2i  
alg=alg2  
clean=.8  
o=\\MyIseries\QDLS\LOGO.afp
```

---

## Error Recovery

The transform can encounter two types of errors; fatal and nonfatal.

When the transform finds a fatal error, the last output page might be incomplete. These are fatal errors:

- Errors in the user-specified parameters; either in the environment variables, the configuration file, or command line arguments. These errors include unrecognized options, incorrect values for parameters, or inability to open the configuration file or the output file.
- Out of memory errors. The requested memory is larger than the memory specified using the **-M** option or the dynamic memory allocation has failed.
- Disk space errors. If the file system is full, the transform deletes the partial output file before terminating. See the **-ms** and **-msf** options for setting the minimum free disk space requirements.
- Internal errors.

When the transform finds a nonfatal error in a file, the transform tries to recover with the next image in the file and issues a warning message. Errors and warnings are output to standard error.

A nonfatal error is made fatal if **-term** is specified. Nonfatal errors include errors that are encountered in the GIF files:

- The transform cannot open the input file.
- Input file is not GIF.
- The input file contains features that cannot be processed.

---

## Limitations

The **gif2afp** transform has these limitations:

- The timing data from GIF animation sequences is ignored.
- Animation sequences are processed as unrelated images.

---

## Acknowledgments

This software is partially based on the Independent JPEG Group's JPEG compression and decompression code.

---

## Chapter 4. Using the JPEG to AFP Transform

This chapter describes how to install and use the **jpeg2afp** command. This command should be run from a DOS window. In this chapter, *command line* refers to the DOS command line. The chapter contains these sections:

- “Syntax”
- “Overview of jpeg2afp”
- “Installing the Image Transforms” on page 57
- “Planning for jpeg2afp” on page 57
- “Parameters” on page 59
- “Transforming JPEG Data to AFP” on page 68
- “Error Recovery” on page 70
- “Limitations” on page 71
- “Acknowledgments” on page 71

---

### Syntax

```
jpeg2afp [-a ImageType] [-alg ProcessingAlgorithms]
[-calib calibration] [-C ConfigurationFile]
[-cmp compression] [-crop CropFactors] [-fit {trim | scale}]
[-force] [-gcorr GrayscaleMappingTable]
[-j ScanOffsetFileName] [-l ImageLength ]
[-M MemoryBound] [-ms space] [-msf SpaceFraction]
[-o OutputFile] [-outbits NumberOfOutputBits]
[-outcolor OutputColorModel] [-p PageRange]
[-pagetype PageType] [-paper PaperSize]
[-r resolution] [-respath ResourceSearchPath]
[-rot rotation] [-scale ImageSize]
[-sgcorr ScannerCorrection]
[-term | -noterm]
[-thresh HalftoneFile] [-v | -nov]
[-w ImageWidth][-wrkdir WorkDirectory]
[-x LeftMargin] [-y TopMargin] [-z]
[file]
```

Figure 13. jpeg2afp command syntax

---

### Overview of jpeg2afp

The JPEG to AFP transform is a command controlled transform that runs on a PC. It converts Joint Photographic Experts Group (JPEG) data to an AFP overlay, page segment, image object, or document, or to a Postscript Level 2 data stream file. Use the pagetype parameter to specify the type of output you want. You can use either of these tools to turn the output AFP resource (overlay or page segment) into a resource on the OS/400:

- iSeries Access. Refer to *AS/400 Guide to Advanced Function Presentation and Print Services Facility* for instructions.
- OS/400 commands. See “Turning a PC AFP Resource into an OS/400 Resource” on page 141.

The JPEG images must be compressed using baseline lossy JPEG compression, and only 8 bits per channel and Huffman coding are supported. The output is bilevel, 4-bit or 8-bit grayscale, or 24-bit YCbCr color. The transform automatically uses halftoning to convert the grayscale and color images to bilevel.

If possible, the transform rewraps the JPEG raster data in MO:DCA-P headers. However, most 24-bit color images are not suitable for rewrapping.

This transform runs on Windows NT service pack 4, Windows 2000, Windows 98 and Windows 95.

## Output Data Types

The output data stream is MO:DCA IS/1, MO:DCA IS/2, or PostScript Level 2. Use the **-a** option to specify the output data stream type. IS/1 images are bilevel and are encoded as IOCA Function Set 10 (FS10) or IM1. IM1 images are uncompressed. IOCA FS10 output images can be uncompressed or compressed by one of the four available compression algorithms (see the **-cmp** option).

IS/2 images can be either bilevel, 4-bit or 8-bit grayscale, or 24 bit YCbCr color. IS/2 images are encoded as IOCA FS11, FS42, or FS45. Multi-bit images are encoded as IOCA FS11 or FS45.

PostScript Level 2 images can be uncompressed or ITU-T T.6 Group 4 bilevel, uncompressed 8-bit grayscale or uncompressed 24-bit RGB color. By default, the transform leaves scaling and halftoning to the printer. That is, color JPEG images are output as 24-bit RGB color images.

If the input image is grayscale or color and the output image type is bilevel, the transform internally scales the image to the desired size first. It then uses a halftoning algorithm to convert it to bilevel.

## Scaling

To maintain image quality, scaling is done before halftoning. The **-alg** option controls scaling algorithms. The size and position of output image depend on several factors:

- The size specified to the transform
- The form map specified at print time
- Forms and options that are loaded in the printer

If the image is too large for page, it is cropped by default. You can override this with the **-fit scale** option. If paper size is not specified, the default paper size is increased to fit the image.

To explicitly specify the image size, use **-scale**. **-scale orig** forces the output image to be the same size as the original image in most cases. An image has to be scaled if the original image is being halftoned or has different resolutions in the X and Y directions (non-square pixels). By default, the transform leaves scaling to the printer. There are two exceptions:

- Images that are being halftoned, which are always scaled in the transform
- Images that have non-square pixels

**Note:** Because JPEG files do not specify resolutions, the image is assumed to have the resolution of the output device.



## Halftoning

When necessary, the transform uses a halftoning algorithm to convert the image to bilevel. When an image is converted to bilevel, the characteristics of the output device such as dot shape and dot gain must be taken into consideration. The internal grayscale mapping table is optimized for the 600-pel IBM Infoprint 4000 laser printer. The **-gcorr** option can be used to specify a different mapping of gray levels.

The **-alg** option controls halftoning algorithms. For very light or dark images, halftoning algorithms can be automatically re-calibrated. Halftoning algorithms are very computation-intensive and slow. You will notice a decrease in performance when images are halftoned.

---

## Installing the Image Transforms

In order to use the JPEG to AFP transform, you must install it on your PC. To install the image transforms, follow these steps:

1. Right-click Network Neighborhood and select **Find Computer....** Specify the name of your OS/400 in the **Named:** field.
2. When Windows finds your OS/400, double-click your system to display the contents.
3. Expand QIBM → ProdData → InfoprintServer → Transforms → Install → Image.
4. Double-click **setup.exe** and follow the directions.

**Note:** Installing the image transforms allows you to use the GIF to AFP, JPEG to AFP, and TIFF to AFP transforms.

---

## Planning for jpeg2afp

Before using the transform, do the following:

- Set up a configuration file - optional. See “Setting up a Configuration File”
- Set up environment variables - optional. See “Using Environment Variables” on page 58
- Edit your Windows path - optional. See “Editing Your Windows Path” on page 41.
- Ensure that you have the proper authorities. See “Authorities” on page 59

## Setting up a Configuration File

The configuration file is one place you can specify options for the transform. Entries are of the form *option=value*, where each pair is on a separate line. Unless otherwise noted, options and other syntax are the same for entries in the configuration file as for the command line arguments. You cannot specify input files in the configuration file.

The following options cannot be used in the configuration file:

- **-calib**
- **-C**

Options without values, such as **-v**, must be given the value **yes**.

Specify the configuration file on the command line with the **-C** option. If you do not specify one, the transform checks for the default configuration file **C:\Program Files\IBM Printing Systems\jpeg2afp\jpeg2afp.cfg** and uses that, if it exists.

The following is an example of a valid configuration file. It specifies that the output image is 8.5 by 11 inches and the resolution of the output device is 300 dpi. The output filename is myfile.afp and it is stored in C:\mydirectory. Verbose mode is on, and the output is a page segment. The spaces surrounding the equals ( = ) sign are optional, but if one is specified after the equals sign on a parameter that specifies a file name, the space is included in the file name.

```
scale = letter
r = 300
o =C:\mydirectory\myfile.afp
v = yes
pagetype = pseg
```

**Note:** Parameters specified in the configuration file are overridden by parameters that are specified on the command line.

## Using Environment Variables

You can use environment variables to specify parameters for the JPEG to AFP transform. Environment variables have the form Variable: JPEG2AFP\_*option*, Value: *value*, where *option* and *value* are the same as the command line and configuration file option and value. If the option does not have a value, specify a blank surrounded by quotation marks ( " ").

You cannot use **-calib** as an environment variable.

To achieve the same effect as the configuration file in the last example, add the following statements to your environment variables on your Windows system:

VARIABLE	VALUE
JPEG2AFP_scale	letter
JPEG2AFP_r	300
JPEG2AFP_o	C:\mydirectory\myfile.afp
JPEG2AFP_v	" "

To edit your Windows environment variables, except with Windows 98, follow these steps:

1. Right-click **My Computer** and select **Properties**.
2. Select the **Environment** page.
3. Specify the parameters you want by repeating these steps as many times as necessary:
  - a. Fill in **Variable** and **Value** for the parameter you want to use.
  - b. Click **Set**.
4. Click **Apply**.
5. Restart your computer for the changes to take effect.

To edit your Windows environment variables for the current session with Windows 98, follow these steps:

1. Open a DOS prompt.
2. Specify the parameters you want by entering this command as many times as necessary: set *variable=value*. For example to set JPEG2AFP\_scale to letter, specify set JPEG2AFP\_scale=letter.
3. Exit the DOS command prompt.

To edit your Windows environment variables for every session, put the lines described in step 2 in your Autoexec.bat file.

**Note:** Environment variables are overridden by parameters that are specified in the configuration file or on the command line.

## Editing Your Windows Path

If you put the directory that contains the transform executable in your Windows path, you can run the transform command from any directory. If you installed the transforms in the default directory and you run the transforms from the C directory, you do not have to edit the path. Follow these steps to edit the path, except with Windows 98.

1. Right-click **My Computer** and select **Properties**.
2. Select the **Environment** page.
3. In the System Variables area, click on the current path.
4. Add the new path to the end in the format `VALUE1;VALUE2;`. For example, if you installed the transforms in the default directory, add this to the end of the current path: `c:\Program Files\IBM Printing Systems\bin;`.
5. Click **Apply**.
6. The changes take effect immediately.

To edit your path with Windows 98, follow these steps:

1. Open `C:\Autoexec.bat` in an editor.
2. Add this line to the end of the file: `SET PATH=directory;`. For example, if you installed the transforms in the default directory, add this to the end of the file: `SET PATH=c:\Program Files\IBM Printing Systems\bin;`.

## Authorities

You need these authorities to use this transform:

- Input file – read authority
- Output file – write authority

---

## Parameters

This section describes how to use the **jpeg2afp** parameters.

**-a {ioca | ioca10 | ioca11 | ioca42 | ioca45 | im1 | PS.2}**

Specifies the output image type. See also **-alg**.

**ioca** The output image type is IOCA FS10.

Make sure that your printer supports IOCA before choosing this value. If your printer does not support IOCA, choose **im1**.

**ioca10**  
The output image type is IOCA FS10.

Make sure that your printer supports IOCA before choosing this value. If your printer does not support IOCA, choose **im1**.

**ioca11**  
The output image type is IOCA FS11. IOCA FS11 cannot be used with an IPDS printer.

**ioca42**  
The output image type is IOCA FS42.

Make sure that your printer supports IOCA before choosing this value. If your printer does not support IOCA, choose **im1**.

## **ioca45**

The output image type is IOCA FS45.

Make sure that your printer supports IOCA before choosing this value. If your printer does not support IOCA, choose **im1**. If you use this value, JPEG is the default output compression type and the image is subsampled to 300 dpi. If **-cmp lzw** is specified, tLZW is used to compress the image and it is not subsampled.

## **im1**    The output image type is IM1.

IM1 images are uncompressed and therefore use more space. The processing might also take longer because IM1 images cannot be scaled by the printer so internal scaling algorithms are used.

## **PS.2**    The output image type is PostScript Level 2.

### **-alg {afp} , {htod1 | htfs} , [ htc11 ]**

Specifies scaling and halftoning algorithms. You can specify multiple values, separated by commas, for one **-alg** option. The effect is the same as specifying multiple **-alg** options. For example, **-alg htfs,htc11** is equivalent to **-alg htfs -alg htc11**.

The following values specify scaling algorithms:

**afp**        Scaling is done by the default algorithm in the printer. This makes the transform faster. If the image must be reduced, however, this algorithm might lose some image information, such as thin lines. This value is the default for bilevel images with IOCA output.

The following values specify halftoning algorithms. Each halftoning algorithm uses a different internal default calibration curve. If automatic calibration is turned on, a different set of calibration curves is used.

**htod1**    Halftoning is done using an ordered dither with a screen derived from the value of the **-thresh** option. The default is an 85 line per inch screen.

**htfs**        Halftoning is done by the Floyd-Steinberg algorithm.

**htc11**    Re-calibrates the halftoning algorithm for each image. **htc11** forces the transform to read the whole image into memory and requires an additional pass through the image. It should be used only for very light or very dark images.

See also **-gcorr**, **-paper**, **-scale**, **-thresh**, **-l**, **-w**, **-x**, and **-y**.

### **-calib {scanner | printer | patch | patchr} , name1...**

Specifies the transform calibration for the scanner or printer. You cannot specify **-calib** in the configuration file or as an environment variable.

**scanner**    The file is calibrated for the scanner. If this value is specified, you must also specify **-sgcorr**.

**printer**    The file is calibrated for the printer.

**patch**        If this value is specified, any value specified for **-sgcorr** is ignored.

**patchr**      If this value is specified, any value specified for **-sgcorr** is ignored.

*name1*        Specifies the name of the file being calibrated. You can specify multiple files.

**-C ConfigurationFileName**

Specifies the name of a configuration file. If multiple **-C** options are given on the command line, they are processed in the order in which you specified them.

The default configuration file is **C:\Program Files\IBM Printing Systems\jpeg2afp\jpeg2afp.cfg**.

*ConfigurationFileName*

The name and location of a configuration file. If there are spaces in the file name, enclose the file name in double quotation marks.

**-cmp {g4|lzw | none | mh | mmr | g3 | jpeg | abic }**

Specifies the output image compression algorithm.

**g4** ITU-T T.6 Group 4 algorithm. This is the default compression for bilevel images. Make sure that your printer supports the Group 4 compression algorithm before specifying this value.

**lzw** Lempel-Zif-Welch algorithm.

**none** The output is uncompressed. This is the default for non-JPEG grayscale and color images.

**mh** ITU-T T.4 Group 3 Modified Huffman algorithm

**mmr** IBM Modified Modified Read algorithm

**g3** ITU-T T.4 Group 3 Modified Read algorithm

**jpeg** JPEG nondifferential Huffman coding with baseline DCT algorithm. This algorithm can only compress 8-bit grayscale and 24-bit color images.

**abic** ABIC algorithm

**-crop t,b,l,r [ d | i | m | p ]**

Specifies how much the image is cropped. Cropping is done in the context of the output page, so that "top" means the top of the output page, regardless of the **-rot** option. The resulting image has the size specified by the **-scale** option. If you want the image cropped from at least one direction, you must specify all of the commas. For example **-crop ,5,,4** and **-crop 0,5,0,4** tell the transform to crop the image five dots from the bottom and four dots from the right. **-crop 5,4** gives you an error. See also **-j**.

The following values specify the amount of cropping:

**t** Specifies how much should be cropped from the top edge. Designate a positive or negative decimal value. A negative number indicates that a blank space is inserted.

**b** Specifies how much should be cropped from the bottom edge. Designate a positive or negative decimal value. A negative number indicates that a blank space is inserted.

**l** Specifies how much should be cropped from the left edge. Designate a positive or negative decimal value. A negative number indicates that a blank space is inserted.

**r** Specifies how much should be cropped from the right edge. Designate a positive or negative decimal value. A negative number indicates that a blank space is inserted.

The following values specify the unit of measurement:

**d** Measurements are in dots (pels).

**i** Measurements are in inches.

**m**      Measurements are in millimeters.

**p**      Measurements are in points.

**-fit {trim | scale}**

Specifies what the printer does when the image is too large to fit on the paper. Specify paper size with the **-paper** option.

**trim**    The image is positioned and then trimmed to make it smaller.

**scale**   The image is scaled to make it smaller.

**-force**

Specifies that **jpeg2afp** decompresses and recompresses the image, regardless of other factors.

**-gcorr *FileName***

Specifies a file that contains a grayscale mapping table for halftoning grayscale and color images. The mapping table must be provided to compensate for the printing characteristics of the particular output device and paper type, such as dot gain. This file must contain one of these:

- 256 real numbers, one for each level of gray in the 8-bit grayscale image
- The PostScript **settransfer** operator

The algorithm converts every image to 8-bit grayscale before applying the halftoning algorithm. The default internal mapping table is optimized for the 600-pel IBM Infoprint 4000 laser printer. Only experienced users should change this. PostScript code submitted by the **-thresh** option can have the same effect as **-gcorr**. See also **-alg** and **-thresh**.

*FileName*      Specifies the name and location of a file that contains a grayscale mapping table. If there are spaces in the file name and you are specifying this value on the command line, enclose the file name in double quotation marks.

**-j *ScanOffsetFileName***

Specifies the name of the scan offset file. Sometimes input images must be shifted to get the desired positioning on the page. The scan offset file contains the directions for shifting such images. The file has this format:

```
[attribute]
values
.
.
.
values
[attribute]values
.
.
.
values
```

These are the recognized attributes:

**[Units]**

Allowed values are: millimeters, **mm**, **inches**, **in**, **points**, **dots**, and **pels**

**[Page\_Offset\_Type]**

Allowed values are: **crop** and **grow**

**[Page\_Offsets]**

The values are in the format *FileName*, **H***horiz*, **V***vert* Each entry refers to an input file. The file names or leading commas must be specified

but are currently ignored. The offset factors are applied to each file in turn. The *horiz* and *vert* values are the amount that the image is shifted in the horizontal and vertical directions. The positive directions are down and right, the origin is in the left upper corner of the page. The shifts are specified context of the output page.

Extra spaces and blank lines are allowed. Unrecognized attributes and their values are ignored.

If the image is moved left or up, part of the bitmap is removed. If the image is moved down or right, removal is governed by the value of the **[Page\_Offset\_Type]** attribute.

**-l {11i | *length*[d | *m* | *i* | *p*]}**

Specifies the paper length. See also **-w** and **-paper**.

11i     The paper is 11 inches long.

*length*   Specifies the length of the paper. Give the length in decimal format.

The following values specify the unit of measurement:

d     The length is measured in dots. This value only uses the whole number part of a decimal value.

*m*     The length is measured in millimeters.

*i*     The length is measured in inches.

*p*     The length is measured in points.

**-lut *FileName***

Specifies the color mapping look up table. The table must be appropriate for the file's color type. For example, if your file is RGB, the look up table must be for RGB mapping.

*FileName*

Specifies the name and location of the file that contains the look up table. If there are spaces in the file name and you are specifying this value on the command line, enclose the file name in double quotation marks.

**-M { 0 | *nnn* }**

Specifies the amount of memory available to **jpeg2afp**.

0     There is no memory limit.

*nnn*   Specifies the maximum amount of memory that is allocated to **jpeg2afp**, in kilobytes. *nnn* is a value in the range of zero to the maximum value that Windows allows.

**-ms {10 | *nnn* }**

Specifies the amount of disk space that must be available on the file system that contains the output file. This limit is enforced every time data is written to the output file. If the output is stdout, this option is ignored. See also **-msf**.

10     There must be at least 10 kilobytes of disk space available on the file system that contains the output file.

*nnn*   There must be at least *nnn* kilobytes of disk space available on the file system that contains the output file.

**-msf {0.1 | *x.xxx* }**

Specifies the minimum amount of available space that must be on the file

system containing the output file. The space is given as a fraction of the total space in the file system. This limit is enforced every time data is written to the output file. If the output is stdout, this option is ignored. The default is **0.1**. See also **-ms**.

**0.1** One-tenth of the total space in the file system must be available.

*x.xxx* Specifies a number in the range of 0-1 that designates the portion of the space that must be available.

**[-noterm | -term]**

Specifies whether error recovery is enabled. When error recovery is enabled, errors in JPEG files are treated as nonfatal. If error recovery is disabled, when the transform finds an error in a JPEG file, the transform ends with an error message and a nonzero return code.

**-noterm** Enables error recovery.

**-term** Disables error recovery.

**[-nov | -v]**

Specifies whether verbose mode is on. When verbose mode is on, the transform sends a message as it opens each resource file. It then echoes the command line and sends a message for each file name as it is processed. When verbose mode is off, the transform sends only error messages.

**-nov** verbose mode is off.

**-v** verbose mode is on.

**-o FileName**

Specifies the output file name. This is optional. If you do not specify an output file name or request standard output, the AFP file is stored in the same location as the input file with the same name as the input file, but with .afp as the extension.

*FileName* The output file name and location. If there are spaces in the file name and you are specifying this value on the command line, enclose the file name in double quotation marks. Specify **-** or **stdout** for stdout. No file is created if you specify stdout.

If you want to use this file on your OS/400 system, specify `\\system\QDLS\file.afp`. *system* is the name of your OS/400 system and *file* is the name of the output file.

For information about importing the AFP file from your PC to your OS/400 system, see "Turning a PC AFP Resource into an OS/400 Resource" on page 141 or refer to for instructions to use iSeries Access.

**-outbits {1 | 24 | 4 | 8 }**

Specifies the number of bits per pel in the output.

**1** The output has 1 bit per pel. If the output image type is IOCA FS10 or IM1, the number of output bits is set to 1 and this option is ignored.

**24** The output has 24 bits per pel. This is the default for IOCA FS11 output. This number represents the maximum number of bits that are used per pel. Thus, even if the number of bits is set to 24, a 4-bit grayscale image is still displayed in 4-bit format.

**4** The output has 4 bits per pel.

**8** The output has 8 bits per pel.



**-outcolor {ycbcr | rgb | ycrCb}**

Specifies the output color model. This value is ignored unless the image type (specified with **-a**) is IOCA FS11 and the number of output bits (specified with **-outbits**) is 24.

**ycbcr** The output color model is YCbCr.

**rgb** The output color model is RGB.

**ycrCb** The output color model is YCrCb.

**-p {even | odd | nn | nn-mm | nn-}**

Specifies which pages the output should contain. If no **-p** option is given, all the pages are output. You can specify **-p** multiple times. Their effect is cumulative. The pages are always printed in ascending order and are numbered starting with one.

**even** Print all even pages.

**odd** Print all odd pages.

**nn** Print page *nn*.

**nn-mm**  
Print pages *nn* to *mm*.

**nn-** Print all pages starting with page *nn*.

**-pagetype {page | overlay | ovly | pseg | object}**

Specifies the output page type. The same page type is set for every page in the output. However, you can use **-p** to output only specified pages. See also **-a**.

**page** All output is pages.

**overlay** All output is overlays.

**ovly** All output is overlays.

**pseg** All output is page segments.

**object** All output is objects.

**-paper {letter | a5 | a4 | a3 | folio | legal | ledger | xxx.xx,yyy.yy[d | m | i | p]}**

Specifies the output paper size. If the paper size is not specified, the paper size is increased, if necessary, to contain the output image.

See also **-l**, **-w**, and **-scale**.

**letter** 8.5 by 11 inches (216 by 279 mm)

**a3** A3 format (297 by 420 mm or 11.69 by 16.54 inches)

**a4** A4 format (210 by 297 mm or 8.27 by 11.69 inches)

**a5** A5 format (148 by 210 mm or 5.83 by 8.27 inches)

**folio** 8.5 by 13.0 inches (216 by 330 mm)

**legal** 8.5 by 14.0 inches (216 by 356 mm)

**ledger** 11.0 by 17.0 inches (279 by 432 mm)

**xxx.xx,yyy.yy [d | i | m | p]**

Specifies the page dimensions.

**xx.xx** Specifies the horizontal page dimension.

**yy.yy** Specifies the vertical page dimension.

The following values specify the unit of measurement:

- d**      Measurements are in dots (pels).
- i**      Measurements are in inches.
- m**      Measurements are in millimeters.
- p**      Measurements are in points.

**-r {300 | *nnn*}**

Specifies the output device resolution in dots per inch.

300      The output device resolution is 300 dots per inch.

*nnn*      The output device resolution is *nnn* dots per inch.

**-respath *directory*[:*directory*...]**

Specifies the search path for resource files, such as the scan offset files and grayscale calibration curve files. If a resource file name is specified as a relative name, the transform searches every directory in the path, in the order specified in the path, until it finds the file or fails. This search path does not apply to anything in the configuration file.

*directory*      Specifies the search path for resource files. The default is **C:\Program Files\IBM Printing Systems\config**.

**-rot {0 | 90 | 180 | 270 [ p | i ]}**

Specifies how far to rotate the image and where the image is rotated.

The following values specify how far to rotate the image:

0      The image is not rotated.

90      The image is rotated clockwise 90 degrees.

180      The image is rotated clockwise 180 degrees.

270      The image is rotated clockwise 270 degrees.

The following values specify where the image is rotated:

p      The printer rotates the image.

i      The transform rotates the image. When i is specified, the transform takes longer to run and uses more memory.

**-scale {orig | a5 | a4 | a3 | letter | folio | legal | ledger | *xxx%* | *xxx.xx,yyy.yy* [d | m | i | lp ]}**

Specifies the output image size. The image is scaled to this size and centered in the area defined by the paper size and margin options.

orig      Preserve the input image size, regardless of the printable paper area. The image still has to be scaled if the input file specifies non-square pels or is halftoned.

a3      A3 format (297 by 420 mm or 11.69 by 16.54 inches)

a4      A4 format (210 by 297 mm or 8.27 by 11.69 inches)

a5      A5 format (148 by 210 mm or 5.83 by 8.27 inches)

letter      8.5 by 11 inches (216 by 279 mm)

folio      8.5 by 13.0 inches (216 by 330 mm)

legal      8.5 by 14.0 inches (216 by 356 mm)

ledger      11.0 by 17.0 inches (279 by 432 mm)

**xx%** Scaling is relative to the original image size. Numbers below 100% mean reduction, while numbers above 100% mean enlargement. For example, **-scale 200%** causes the output image to be twice as long and twice as wide as the input image.

**xx.xx,yy.yy[d | m | i | p]**

Specifies the image dimensions.

**xx.xx** Specifies the horizontal dimension.

**yy.yy** Specifies the vertical dimension.

The following values specify the unit of measurement:

**d** Measurements are in dots (pels).

**i** Measurements are in inches.

**m** Measurements are in millimeters.

**p** Measurements are in points.

**-sgcorr** *FileName*

Specifies the name of the scanner-related calibration file. If **-calib** is not specified, this option is ignored.

*FileName* The name and location of the scanner calibration file. If there are spaces in the file name and you are specifying this value on the command line, enclose the file name in double quotation marks.

**-term**

See **[-noterm | -term]**

**-thresh** *FileName*

Specifies a file that contains a PostScript Type 1 or Type 3 halftone dictionary. Alternatively, the PostScript code can specify the **setscreen** operator instead of a Type 1 dictionary. The halftone cell in the dictionary is used to overwrite the default ordered dither clustered dot halftone cell. If the PostScript code contains a transfer function, either in the halftone dictionary or specified by the **settransfer** operator, the current grayscale correction curve is overwritten as if **-gcorr** were used.

*FileName* The name and location of the file that contains an appropriate halftone dictionary. If there are spaces in the file name and you are specifying this value on the command line, enclose the file name in double quotation marks.

**-v** See **[-nov | -v]**

**-w {8.5i | yyy.yy[d | m | i | p]}**

Specifies the paper width. See also **-l** to set the paper length or **-paper** to set both dimensions at the same time.

**8.5i** The page width is 8.5 inches.

The following values specify the unit of measurement:

**d** The width is measured in dots (pels).

**i** The width is measured in inches.

**m** The width is measured in millimeters.

**p** The width is measured in points.

**-wrkdir** *WorkDirectoryName*

Input to **jpeg2afp** from stdin must be cached before it can be processed. This option specifies which directory to use for caching the input. If this option is not specified, or the directory cannot be opened, the transform tries to use **C:\Program Files\IBM Printing Systems\jpeg2afp** as the working directory. If the transform cannot use **C:\Program Files\IBM Printing Systems\jpeg2afp**, it uses **C:\temp**.

The stdin cache file is not visible to the user because it is immediately unlinked after creation. Thus, the cache file is always deleted when **jpeg2afp** finishes, even if it ends abnormally.

**-x {0 | *nnn.nn*[*dl m i l p*]}**

Specifies the left margin.

**0**           There is no left margin.

*nnn.nn*  
              The left margin is *nnn.nn* units.

The following values specify the unit of measurement:

**d**           The margin is measured in dots (pels).

**i**           The margin is measured in inches.

**m**           The margin is measured in millimeters.

**p**           The margin is measured in points.

**-y {0 | *nnn.nn*[*dl m i l p*]}**

Specifies the top margin.

**0**           There is no top margin.

*nnn.nn*  
              The top margin is *nnn.nn* units.

The following values specify the unit of measurement:

**d**           The margin is measured in dots (pels).

**i**           The margin is measured in inches.

**m**           The margin is measured in millimeters.

**p**           The margin is measured in points.

---

## Transforming JPEG Data to AFP

This section describes how to transform JPEG data to AFP and gives examples.

### Input to the JPEG to AFP Transform

Input can be a JPEG file submitted by standard input (stdin) or specified on the command line. If no input file is specified, stdin is assumed. The extension .jpg, .jpeg, .JPG, or .JPEG does not need to be given explicitly. **jpeg2afp** first tries to open the file as specified and, if unsuccessful, tries to append the extensions .jpg, .jpeg, .JPG, and .JPEG in turn.

### Specifying the Output File

The output file name and location can be derived from the input file name and location, specified on **-o**. Alternatively, you can force output to be standard output

(stdout - this is typically output to your screen) by specifying **-o** - or **-o stdout** on the command line or **o = -** or **o = stdout** in the configuration file. If you request standard output, no output file is generated. By default, the output file has the same name as the input file and is stored in the same location as the input file.

If the output file name is not specified explicitly, the transform strips any extension from the input file name and appends the appropriate extension to create the output file name.

For example, all the following commands have `myfile.afp` as the output file:

```
jpeg2afp myfile
jpeg2afp myfile.jpeg
jpeg2afp myfile -o myfile.afp
```

Note that there is no requirement for the explicitly specified input and output files to have `.jpeg` and `.afp` extensions. To process JPEG file `foo.bar` into an AFP file named `foo.bar2`, use this command to invoke the transform:

```
jpeg2afp foo.bar -o foo.bar2
```

## Invoking jpeg2afp

To invoke **jpeg2afp**, go to a DOS prompt and specify `jpeg2afp file -option value...` *file* is the location and name of the JPEG file and *option value* is any parameter you want to specify on the command line. You do not have to specify any parameters on the command line. The parameter values are determined by the following hierarchy (least significant first):

1. Internal defaults
2. Environment variables
3. Configuration file
4. Command-line arguments

### Notes:

1. If there are spaces in the path or name, enclose the string in double quotation marks. For example, "C:\Program Files\IBM Printing Systems\files\file.jpeg".
2. If the directory that contains **jpeg2afp.exe** is not in your PATH specification, you must be in that directory to run the command. If you accepted the default directories at install time, this directory is **C:\Program Files\IBM Printing Systems\bin**. See "Editing Your Windows Path" on page 59 for instructions to change your PATH.

## Command Line Arguments

When using the command line, note the following:

- Options and input file names can be specified in any order.
- Only file names are case-sensitive.
- If an option is specified multiple times, only the last one is recognized.
- If multiple configuration files are given, they are processed in the order in which you specify them.

## Examples

### Example 1

This example transforms a JPEG file called `MyOverlay.jpeg` into an AFP overlay called `MyOverlay.afp`. The output file name is not specified because it is given the

same name as the input file with the .afp extension by default. It is stored in the specified directory so it can easily be imported to the system MyIseries. Enter this command at a DOS prompt:

```
jpeg2afp MyOverlay.jpeg -pagetype overlay -o \\MyIseries\QDLS\
```

The same results can be achieved with this command and configuration file:

```
jpeg2afp MyJpegFile.jpeg -C ConfigFile1
```

ConfigFile1 contains this:

```
pagetype=overlay  
o=\\MyIseries\QDLS\
```

## Example 2

This example transforms a JPEG file called BigLogo.jpeg into an AFP page segment called LOGO. The logo needs to be scaled down and rotated 90 degrees. The output is stored in the specified directory so it can easily be imported to the system MyIseries. Enter this command in a single line at a DOS prompt:

```
jpeg2afp BigLogo.jpeg -pagetype pseg -scale 2,2i -rot 90 -o \\MyIseries\QDLS\LOGO.afp
```

The same results can be achieved with this command and configuration file:

```
jpeg2afp BigLogo.jpeg -C ConfigFile2
```

ConfigFile2 contains this:

```
pagetype=pseg  
scale=2,2i  
alg=alg2  
clean=.8  
o=\\MyIseries\QDLS\LOGO.afp
```

---

## Error Recovery

The transform can encounter two types of errors; fatal and nonfatal.

When the transform finds a fatal error, the last output page might be incomplete. These are fatal errors:

- Errors in the user-specified parameters; either in the environment variables, the configuration file, or command line arguments. These errors include unrecognized options, incorrect values for parameters, or inability to open the configuration file or the output file.
- Out of memory errors. The requested memory is larger than the memory specified using the **-M** option or the dynamic memory allocation has failed.
- Disk space errors. If the file system is full, the transform deletes the partial output file before terminating. See the **-ms** and **-msf** options for setting the minimum free disk space requirements.
- Internal errors.

When the transform finds a nonfatal error in a file, a warning message is issued. Errors and warnings are output to standard error.

A nonfatal error is made fatal if **-term** is specified. Nonfatal errors include errors that are encountered in the JPEG files:

- The transform cannot open the input file.
- The input file is not JPEG.

- The input file contains features that cannot be processed.

---

## Limitations

The **jpeg2afp** transform has these limitations:

- Only Huffman coding with baseline DCT JPEG algorithm is supported.
- At most 8-bit grayscale and 24-bit color are supported.

---

## Acknowledgments

This software is partially based on the Independent JPEG Group's JPEG compression and decompression code.





---

## Chapter 5. Using the TIFF to AFP Transform

This chapter describes how to install and use the **tiff2afp** command. This command should be run from a DOS window. In this chapter, *command line* refers to the DOS command line. The chapter contains these sections:

- “Syntax”
- “Overview of tiff2afp”
- “Installing the Image Transforms” on page 75
- “Planning for tiff2afp” on page 75
- “Parameters” on page 77
- “Transforming TIFF Data to AFP” on page 88
- “Error Recovery” on page 90
- “Limitations” on page 90
- “Acknowledgments” on page 91

---

### Syntax

```
tiff2afp [-a ImageType] [-alg ProcessingAlgorithms]  
[-calib calibration][-choice ImageChoice]  
[-C ConfigurationFile] [-clean cleanup]  
[-cmp compression] [-crop CropFactors]  
[-fit {trim | scale}] [-force]  
[-gcorr GrayscaleMappingTable] [-ink color] [-inv]  
[-j ScanOffsetFileName] [-l ImageLength ]  
[-M MemoryBound] [-mp | -nomp]  
[-ms space] [-msf SpaceFraction]  
[-o OutputFile] [-outbits NumberOfOutputBits]  
[-outcolor OutputColorModel] [-p PageRange]  
[-pagetype PageType] [-paper PaperSize]  
[-r resolution] [-respath ResourceSearchPath]  
[-rot rotation] [-scale ImageSize]  
[-sgcorr ScannerCorrection]  
[-sniff | -nosniff] [-term | -noterm]  
[-thresh HalftoneFile] [-v | -nov]  
[-w ImageWidth] [-wrkdir WorkDirectory]  
[-x LeftMargin] [-y TopMargin] [-z]  
[file]
```

Figure 14. tiff2afp command syntax

---

### Overview of tiff2afp

The TIFF to AFP transform is a command controlled transform that converts tagged image file format (TIFF) data to an AFP overlay, page segment, image object, or document, or to a Postscript Level 2 data stream file. Use the pagetype parameter to specify the type of output you want. You can use either of these tools to turn the output AFP resource (overlay or page segment) into a resource on the OS/400:

- iSeries Access. Refer to *AS/400 Guide to Advanced Function Presentation and Print Services Facility* for instructions.

- OS/400 commands. See “Turning a PC AFP Resource into an OS/400 Resource” on page 141.

The transform processes bilevel (black and white), grayscale and color extended TIFF Revision 6.0 images. TIFF files can contain multiple images. The output is bilevel, 4-bit or 8-bit grayscale, or 24-bit YCbCr color. The transform automatically uses halftoning to convert the grayscale and color images to bilevel. If page numbers are available in the TIFF data, the pages are sorted in this order. The pages are not sorted across file boundaries.

TIFF images are often divided into many segments for efficient memory handling. Some printers terminate the printing in error if a page is too complex. For this reason, `tiff2afp` decompresses, re-partitions, and compresses the data if it deems that there are too many segments in the image.

This transform runs on Windows NT service pack 4, Windows 2000, Windows 98 and Windows 95.

## Output Data Types

The output data stream is MO:DCA IS/1, MO:DCA IS/2, or PostScript Level 2. Use the **-a** option to specify the output data stream type. IS/1 images are bilevel and are encoded as IOCA Function Set (FS) 10 or IM1. IM1 images are uncompressed. IOCA FS10 output images can be uncompressed or compressed by one of the four available compression algorithms (see **-cmp** ).

IS/2 images can be either bilevel, 4-bit or 8-bit grayscale, or 24 bit YCbCr color. IS/2 images are encoded as IOCA FS11, FS42, or FS45. Multi-bit images are encoded as IOCA FS11 or FS45.

PostScript Level 2 images can be uncompressed or ITU-T T.6 Group 4 bilevel, uncompressed 8-bit grayscale or uncompressed 24-bit RGB color. By default, the transform leaves scaling and halftoning to the printer. That is, color TIFF images are output as 24-bit RGB color images.

If the input image is grayscale or color and the output image type is bilevel, the transform internally scales the image to the desired size first. It then uses a halftoning algorithm to convert it to bilevel.

## Scaling

To maintain image quality, scaling is done before halftoning. The options **-alg**, **-clean**, and **-ink** control scaling algorithms. The size and position of output image depend on several factors:

- The size specified to the transform
- The form map specified at print time
- Forms and options that are loaded in the printer

If the image is too large for page, it is cropped by default. You can override this with the **-fit scale** option. If paper size is not specified, the default paper size is increased to fit the image.

To explicitly specify the image size, use **-scale**. **-scale orig** forces the output image to be the same size as the original image in most cases. An image has to be scaled

if the original image is being halftoned or has different resolutions in the X and Y directions (non-square pixels). By default, the transform leaves scaling to the printer. There are two exceptions:

- Images that are being halftoned, which are always scaled in the transform
- Images that have non-square pixels

**Note:** Because TIFF files do not specify resolutions, the image is assumed to have the resolution of the output device.

## Halftoning

When necessary, the transform uses a halftoning algorithm to convert the image to bilevel. When an image is converted to bilevel, the characteristics of the output device such as dot shape and dot gain must be taken into consideration. The internal grayscale mapping table is optimized for the 600-pel IBM Infoprint 4000 laser printer. The **-gcorr** option can be used to specify a different mapping of gray levels.

The **-alg** option controls halftoning algorithms. For very light or dark images, halftoning algorithms can be automatically re-calibrated. Halftoning algorithms are very computation-intensive and slow. You will notice a decrease in performance when images are halftoned.

---

## Installing the Image Transforms

In order to use the TIFF to AFP transform, you must install it on your PC. To install the image transforms, follow these steps:

1. Right-click Network Neighborhood and select **Find Computer...** Specify the name of your OS/400 in the **Named:** field.
2. When Windows finds your OS/400, double-click your system to display the contents.
3. Expand QIBM → ProdData → InfoprintServer → Transforms → Install → Image.
4. Double-click **setup.exe** and follow the directions.

**Note:** Installing the image transforms allows you to use the GIF to AFP, JPEG to AFP, and TIFF to AFP transforms.

---

## Planning for tiff2afp

Before using the transform, do the following:

- Set up a configuration file - optional. See “Setting up a Configuration File”.
- Set up environment variables - optional. See “Using Environment Variables” on page 76.
- Edit your Windows path - optional. See “Editing Your Windows Path” on page 77.
- Ensure that you have the proper authorities. See “Authorities” on page 77.

## Setting up a Configuration File

The configuration file is one place you can specify options for the transform. Entries are of the form *option=value*, where each pair is on a separate line. Unless otherwise noted, options and other syntax are the same for entries in the configuration file as for the command line arguments. You cannot specify input files in the configuration file.

The following options cannot be used in the configuration file:

- **-calib**
- **-C**

Options without values, such as **-v**, must be given the value **yes**.

Specify the configuration file on the command line with the **-C** option. If you do not specify one, the transform checks for the default configuration file **C:\Program Files\IBM Printing Systems\tiff2afp\tiff2afp.cfg** and uses that, if it exists.

The following is an example of a valid configuration file. It specifies that the output image is 8.5 by 11 inches and the resolution of the output device is 300 dpi. The output filename is myfile.afp and it is stored in C:\mydirectory. Verbose mode is on, and the output is a page segment. The spaces surrounding the equals (=) sign are optional, but if one is specified after the equals sign on a parameter that specifies a file name, the space is included in the file name.

```
scale = letter
r = 300
o =C:\mydirectory\myfile.afp
v = yes
pagetype = pseg
```

**Note:** Parameters specified in the configuration file are overridden by parameters that are specified on the command line.

## Using Environment Variables

You can use environment variables to specify parameters for the TIFF to AFP transform. Environment variables have the form Variable: TIFF2AFP\_*option*, Value: *value*, where *option* and *value* are the same as the command line and configuration file option and value. If the option does not have a value, specify a blank surrounded by quotation marks (" ").

You cannot use **-calib** as an environment variable.

To achieve the same effect as the configuration file in the last example, add the following statements to your environment variables on your Windows system:

VARIABLE	VALUE
TIFF2AFP_scale	letter
TIFF2AFP_v	" "

To edit your Windows environment variables, except with Windows 98, follow these steps:

1. Right-click **My Computer** and select **Properties**.
2. Select the **Environment** page.
3. Specify the parameters you want by repeating these steps as many times as necessary:
  - a. Fill in **Variable** and **Value** for the parameter you want to use.
  - b. Click **Set**.
4. Click **Apply**.
5. Restart your computer for the changes to take effect.

To edit your Windows environment variables for the current session with Windows 98, follow these steps:

1. Open a DOS prompt.

2. Specify the parameters you want by entering this command as many times as necessary: `set variable=value`. For example to set `TIFF2AFP_scale` to letter, specify `set TIFF2AFP_scale=letter`.
3. Exit the DOS command prompt.

To edit your Windows environment variables for every session, put the lines described in step 2 in your `Autoexec.bat` file.

**Note:** Environment variables are overridden by parameters that are specified in the configuration file or on the command line.

## Editing Your Windows Path

If you put the directory that contains the transform executable in your Windows path, you can run the transform command from any directory. If you installed the transforms in the default directory and you run the transforms from the C directory, you do not have to edit the path. Follow these steps to edit the path, except with Windows 98.

1. Right-click **My Computer** and select **Properties**.
2. Select the **Environment** page.
3. In the System Variables area, click on the current path.
4. Add the new path to the end in the format `VALUE1;VALUE2;`. For example, if you installed the transforms in the default directory, add this to the end of the current path: `c:\Program Files\IBM Printing Systems\bin;`.
5. Click **Apply**.
6. The changes take effect immediately.

To edit your path with Windows 98, follow these steps:

1. Open `C:\Autoexec.bat` in an editor.
2. Add this line to the end of the file: `SET PATH=directory;`. For example, if you installed the transforms in the default directory, add this to the end of the file: `SET PATH=c:\Program Files\IBM Printing Systems\bin;`.

## Authorities

You need these authorities to use this transform:

- Input file – read authority
- Output file – write authority

---

## Parameters

This section describes how to use the **tiff2afp** parameters.

**-a {ioca | ioca10 | ioca11 | ioca42 | ioca45 | im1 | PS.2}**

Specifies the output image type. See also **-alg**.

**ioca** The output image type is IOCA FS10.

Make sure that your printer supports IOCA before choosing this value. If your printer does not support IOCA, choose **im1**.

**ioca10**

The output image type is IOCA FS10.

Make sure that your printer supports IOCA before choosing this value. If your printer does not support IOCA, choose **im1**.

**ioca11**

The output image type is IOCA FS11. IOCA FS11 cannot be used with an IPDS printer.

**ioca42**

The output image type is IOCA FS42.

Make sure that your printer supports IOCA before choosing this value. If your printer does not support IOCA, choose **im1**.

**ioca45**

The output image type is IOCA FS45.

Make sure that your printer supports IOCA before choosing this value. If your printer does not support IOCA, choose **im1**. If you use this value, JPEG is the default output compression type and the image is subsampled to 300 dpi. If **-cmp lzw** is specified, tLZW is used to compress the image and it is not subsampled.

**im1**

The output image type is IM1.

IM1 images are uncompressed and therefore use more space. The processing might also take longer because IM1 images cannot be scaled by the printer and internal scaling algorithms are used.

**PS.2** The output image type is PostScript Level 2.

**-alg {afp | alg1 | alg2 | alg3} , {htod1 | htfs} , [ htcal1 ]**

Specifies scaling and halftoning algorithms. You can specify multiple values, separated by commas, for one **-alg** option. The effect is the same as specifying multiple **-alg** options. For example, **-alg htfs,htcal1** is equivalent to **-alg htfs -alg htcal1**.

The following values specify scaling algorithms:

**afp**

Scaling is done by the default algorithm in the printer. This makes the transform faster. If the image must be reduced, however, this algorithm might lose some image information, such as thin lines. This value is the default for bilevel images with IOCA output.

**alg1**

Scaling is done by the internal scaling algorithm. This algorithm will not discard any ink in the image. White space in the image, however, might be lost. This algorithm makes the transform run longer.

**alg2**

Scaling is done by a more flexible version of **alg1**. If this value is selected, **-clean** can be used to specify the amount of ink cleanup that is performed.

**alg3**

Scaling is done by a scaling algorithm that works by deleting or duplicating rows and columns in the image. This is a fast general-purpose algorithm. This value is the default for bilevel images with IM1 output.

The following values specify halftoning algorithms. Each halftoning algorithm uses a different internal default calibration curve. If automatic calibration is turned on, a different set of calibration curves is used.

**htod1**

Halftoning is done using an ordered dither with a screen derived from the value of the **-thresh** option. The default is an 85 line per inch screen.

**htfs**

Halftoning is done by the Floyd-Steinberg algorithm.

**htcal1** Re-calibrates the halftoning algorithm for each image. **htcal1** forces the

transform to read the whole image into memory and requires an additional pass through the image. It should be used only for very light or very dark images.

See also **-clean**, **-gcorr**, **-ink**, **-paper**, **-scale**, **-thresh**, **-l**, **-w**, **-x**, and **-y**.

**-calib {scanner | printer | patch | patchr} , name1...**

Specifies the transform calibration for the scanner or printer. You cannot specify **-calib** in the configuration file or as an environment variable.

**scanner** The file is calibrated for the scanner. If this value is specified, you must also specify **-sgcorr**.

**printer** The file is calibrated for the printer.

**patch** If this value is specified, any value specified for **-sgcorr** is ignored.

**patchr** If this value is specified, any value specified for **-sgcorr** is ignored.

*name1* Specifies the name of the file being calibrated. You can specify multiple files.

**-choice {full | reduced | both }**

TIFF image files can contain both full and reduced resolution versions of each image. This option specifies which type of image the transform outputs. See also **-mp** and **-nomp**

**full** The transform outputs the full resolution images only.

**reduced** The transform outputs the reduced resolution images only.

**both** The transform outputs both the full and reduced resolution images.

**-C ConfigurationFileName**

Specifies the name of a configuration file. If multiple **-C** options are given on the command line, they are processed in the order in which you specified them.

The default configuration file is **C:\Program Files\IBM Printing Systems\tiff2afp\tiff2afp.cfg**.

*ConfigurationFileName*

The name and location of a configuration file. If there are spaces in the file name, enclose the file name in double quotation marks.

**-clean {0.5 | x.xxx}**

Specifies when to turn on a pel. The scaled pel is turned on if the specified part of its area, defined by the threshold, is painted in the original image. The threshold is the value **0.5** or *x.xxx*. It is a decimal that defines a fraction. If the fraction of a pel that is painted in the original image is greater than this threshold, the pel is turned on. For example, if you specify **-clean .4**, a pel is turned on any time more than 4 tenths, or 20%, of the pel is painted in the original image. This option is only used if **-alg alg2** is specified.

**0.5** The threshold is .5. **alg2** turns on a pel any time more than half of it is painted in the original image.

*x.xxx* The threshold is *x.xxx*. *x.xxx* is a number in the range of 0-1.

If **-clean 0.0** is specified, **alg2** turns on a pel any time any part of the pel is painted in the original image. Therefore, it behaves like **alg1**.

**-clean 1.0** turns on a pel when more than 100% of the pel is painted in the original image; therefore, it results in a blank image.



**-cmp {g4|none| lzw | mh | mmr | g3 | jpeg | abic }**

Specifies the output image compression algorithm.

**g4** ITU-T T.6 Group 4 algorithm. This is the default compression for bilevel images. Make sure that your printer supports the Group 4 compression algorithm before specifying this value.

**lzw** Lempel-Zif-Welch algorithm.

**none** The output is uncompressed. This is the default for non-JPEG grayscale and color images.

**mh** ITU-T T.4 Group 3 Modified Huffman algorithm

**mmr** IBM Modified Modified Read algorithm

**g3** ITU-T T.4 Group 3 Modified Read algorithm

**jpeg** JPEG nondifferential Huffman coding with baseline DCT algorithm. This algorithm can only compress 8-bit grayscale and 24-bit color images.

**abic** ABIC algorithm

**-crop *t,b,l,r* [ d | i | m | p ]**

Specifies how much the image is cropped. Cropping is done in the context of the output page, so that “top” means the top of the output page, regardless of the **-rot** option. The resulting image has the size specified by the **-scale** option. If you want the image cropped from at least one direction, you must specify all of the commas. For example **-crop ,5,,4** and **-crop 0,5,0,4** tell the transform to crop the image five dots from the bottom and four dots from the right. **-crop 5,4** gives you an error. See also **-j**.

The following values specify the amount of cropping:

***t*** Specifies how much should be cropped from the top edge. Designate a positive or negative decimal value. A negative number indicates that a blank space is inserted.

***b*** Specifies how much should be cropped from the bottom edge. Designate a positive or negative decimal value. A negative number indicates that a blank space is inserted.

***l*** Specifies how much should be cropped from the left edge. Designate a positive or negative decimal value. A negative number indicates that a blank space is inserted.

***r*** Specifies how much should be cropped from the right edge. Designate a positive or negative decimal value. A negative number indicates that a blank space is inserted.

The following values specify the unit of measurement:

**d** Measurements are in dots (pels).

**i** Measurements are in inches.

**m** Measurements are in millimeters.

**p** Measurements are in points.

**-fit {trim | scale}**

Specifies what the printer does when the image is too large to fit on the paper. Specify paper size with the **-paper** option.

**trim** The image is positioned and then trimmed to make it smaller.



**scale** The image is scaled to make it smaller.

**-force**

Specifies that **tiff2afp** decompresses and recompresses the image, regardless of other factors.

**-gcorr** *FileName*

Specifies a file that contains a grayscale mapping table for halftoning grayscale and color images. The mapping table must be provided to compensate for the printing characteristics of the particular output device and paper type, such as dot gain. This file must contain one of these:

- 256 real numbers, one for each level of gray in the 8-bit grayscale image
- The PostScript **settransfer** operator

The algorithm converts every image to 8-bit grayscale before applying the halftoning algorithm. The default internal mapping table is optimized for the 600-pel IBM Infoprint 4000 laser printer. Only experienced users should change this. PostScript code submitted by the **-thresh** option can have the same effect as **-gcorr**. See also **-alg** and **-thresh**.

*FileName* Specifies the name and location of a file that contains a grayscale mapping table. If there are spaces in the file name and you are specifying this value on the command line, enclose the file name in double quotation marks.

**-ink { black | white }**

Specifies whether the transform preserves black or white pels when compressing. The internal scaling algorithm must know which, because “ink” is preserved at the expense of “non-ink” This option is used only with scaling algorithm **alg1** or **alg2**.

If the option **-inv** is used to print the reversed image, the inversion is done as the data is read by the transform, before any processing is done. Therefore, black or white must be specified in the terms of the printed image, not in the terms of the input TIFF image.

**black** The image is black on white.

**white** The image is white on black.

**-inv**

Reverses the image. Areas that are black in the original image become white and vice versa. Note that the reversed image might be black on white or white on black, depending on the original image.

**-j** *ScanOffsetFileName*

Specifies the name of the scan offset file. Sometimes input images must be shifted to get the desired positioning on the page. The scan offset file contains the directions for shifting such images. The file has this format:

```
[attribute]
values
.
.
.
values
[attribute]
values
.
.
.
values
```

These are the recognized attributes:

**[Units]**

Allowed values are: millimeters, **mm**, **inches**, **in**, **points**, **dots**, and **pels**

**[Page\_Offset\_Type]**

Allowed values are: **crop** and **grow**

**[Page\_Offsets]**

The values are in the format *FileName*, **H***horiz*, **V***vert* Each entry refers to an input file. The file names or leading commas must be specified but are currently ignored. The offset factors are applied to each file in turn. The same factors are applied to each image in the file. The *horiz* and *vert* values are the amount that the image is shifted in the horizontal and vertical directions. The positive directions are down and right, the origin is in the left upper corner of the page. The shifts are specified context of the output page.

Extra spaces and blank lines are allowed. Unrecognized attributes and their values are ignored.

If the image is moved left or up, part of the bitmap is removed. If the image is moved down or right, removal is governed by the value of the **[Page\_Offset\_Type]** attribute.

**-l {11i | length[d | m | i | p]}**

Specifies the paper length. See also **-w** and **-paper**.

**11i** The paper is 11 inches long.

*length* Specifies the length of the paper. Give the length in decimal format.

The following values specify the unit of measurement:

**d** The length is measured in dots. This value only uses the whole number part of a decimal value.

**m** The length is measured in millimeters.

**i** The length is measured in inches.

**p** The length is measured in points.

**-lut *FileName***

Specifies the color mapping look up table. The table must be appropriate for the file's color type. For example, if your file is RGB, the look up table must be for RGB mapping.

*FileName*

Specifies the name and location of the file that contains the look up table. If there are spaces in the file name and you are specifying this value on the command line, enclose the file name in double quotation marks.

**-M { 0 | nnn }**

Specifies the amount of memory available to **tiff2afp**.

**0** There is no memory limit.

*nnn* Specifies the maximum amount of memory allocated to **tiff2afp**, in kilobytes. *nnn* is a value in the range of zero to the maximum value that Windows allows.

**[ -mp | -nomp ]**

Each TIFF image is supposed to have a marker bit set if there are multiple pages in the file. This marker is sometimes missing, even if there are multiple pages in the file. This option specifies whether the marker is present.

**-mp** The transform behaves as if the marker is set. The transform processes all the pages in the file, regardless of the marker bit.

**-nomp** The transform looks for the marker bit. All the pages of multi-page TIFF file are processed only if the marker bit is set in the file. If there is a multi-page TIFF file without the marker bit set, only the first page is processed.

**-ms {10 | nnn}**

Specifies the amount of disk space that must be available on the file system that contains the output file. This limit is enforced every time data is written to the output file. If the output is stdout, this option is ignored. See also **-msf**.

**10** There must be at least 10 kilobytes of disk space available on the file system that contains the output file.

**nnn** There must be at least *nnn* kilobytes of disk space available on the file system that contains the output file.

**-msf {0.1 | x.xxx}**

Specifies the minimum amount of available space that must be on the file system containing the output file. The space is given as a fraction of the total space in the file system. This limit is enforced every time data is written to the output file. If the output is stdout, this option is ignored. The default is **0.1**. See also **-ms**.

**0.1** One-tenth of the total space in the file system must be available.

**x.xxx** Specifies a number in the range of 0-1 that designates the portion of the space that must be available.

**[ -nomp ]**

See **[ -mp | -nomp ]**

**[ -nosniff | -sniff ]**

Specifies whether bilevel images are automatically converted to black on white. The conversion algorithm determines the number of black and white bits. If there are fewer white bits, it assumes that the image is white on black, and inverts the image to be black on white. This algorithm is run after the conversion specified by the Photometric Interpretation tag is applied (if any).

This option is ignored if the image is not bilevel.

**-nosniff** Bilevel images are not converted to black on white.

**-sniff** Bilevel images are converted to black on white.

**[ -noterm | -term ]**

Specifies whether error recovery is enabled. When error recovery is enabled, errors in TIFF files are treated as nonfatal. The transform tries to recover with the next image in the file. If error recovery is disabled, when the transform finds an error in a TIFF file, the transform ends with an error message and a nonzero return code.

**-noterm** Enables error recovery.

**-term** Disables error recovery.

**[-nov | -v]**

Specifies whether verbose mode is on. When verbose mode is on, the transform sends a message as it opens each resource file. It then echoes the command line and sends a message for each file name as it is processed. When verbose mode is off, the transform sends only error messages.

**-nov**     verbose mode is off.

**-v**        verbose mode is on.

**-o FileName**

Specifies the output file name. This is optional. If you do not specify an output file name or request standard output, the AFP file is stored in the same location as the input file with the same name as the input file, but with .afp as the extension.

**FileName**     The output file name and location. If there are spaces in the file name and you are specifying this value on the command line, enclose the file name in double quotation marks. Specify **-** or **stdout** for stdout. No file is created if you specify stdout.

If you want to use this file on your OS/400 system, specify `\\system\QDLS\file.afp`. *system* is the name of your OS/400 system and *file* is the name of the output file.

For information about importing the AFP file from your PC to your OS/400 system, see "Turning a PC AFP Resource into an OS/400 Resource" on page 141 or refer to for instructions to use iSeries Access.

**-outbits {1 | 24 | 4 | 8 }**

Specifies the number of bits per pel in the output.

**1**        The output has 1 bit per pel. If the output image type is IOCA FS10 or IM1, the number of output bits is set to 1 and this option is ignored.

**24**        The output has 24 bits per pel. This is the default for IOCA FS11 output. This number represents the maximum number of bits that are used per pel. Thus, even if the number of bits is set to 24, a 4-bit grayscale image is still displayed in 4-bit format.

**4**        The output has 4 bits per pel.

**8**        The output has 8 bits per pel.

**-outcolor {ycbcr | rgb | ycrCb}**

Specifies the output color model. This value is ignored unless the image type (specified with **-a**) is IOCA FS11 and the number of output bits (specified with **-outbits**) is 24.

**ycbcr**     The output color model is YCbCr.

**rgb**        The output color model is RGB.

**ycrCb**     The output color model is YCrCb.

**-p {even | odd | nn | nn-mm | nn-}**

Specifies which pages the output should contain. If no **-p** option is given, all the pages are output. You can specify **-p** multiple times. Their effect is cumulative. The pages are always printed in ascending order and are numbered starting with one.

**even**     Print all even pages.

**odd**       Print all odd pages.

*nn* Print page *nn*.

*nn-mm*  
Print pages *nn* to *mm*.

*nn-* Print all pages starting with page *nn*.

**-pagetype {page | overlay | ovly | pseg | object}**

Specifies the output page type. The same page type is set for every page in the output. However, you can use **-p** to output only specified pages. See also **-a**.

**page** All output is pages.

**overlay** All output is overlays.

**ovly** All output is overlays.

**pseg** All output is page segments.

**object** All output is objects.

**-paper {letter | a5 | a4 | a3 | folio | legal | ledger | *xxx.xx,yyy.yy*[d | m | i | p]}**

Specifies the output paper size. If the paper size is not specified, the paper size is increased, if necessary, to contain the output image.

See also **-l**, **-w**, and **-scale**.

**letter** 8.5 by 11 inches (216 by 279 mm)

**a3** A3 format (297 by 420 mm or 11.69 by 16.54 inches)

**a4** A4 format (210 by 297 mm or 8.27 by 11.69 inches)

**a5** A5 format (148 by 210 mm or 5.83 by 8.27 inches)

**folio** 8.5 by 13.0 inches (216 by 330 mm)

**legal** 8.5 by 14.0 inches (216 by 356 mm)

**ledger** 11.0 by 17.0 inches (279 by 432 mm)

*xx.xx,yy.yy* [d | i | m | p]

Specifies the page dimensions.

*xx.xx* Specifies the horizontal page dimension.

*yy.yy* Specifies the vertical page dimension.

The following values specify the unit of measurement:

**d** Measurements are in dots (pels).

**i** Measurements are in inches.

**m** Measurements are in millimeters.

**p** Measurements are in points.

**-r {300 | *nnn*}**

Specifies the output device resolution in dots per inch.

**300** The output device resolution is 300 dots per inch.

*nnn* The output device resolution is *nnn* dots per inch.

**-respath *directory*[:*directory*...]**

Specifies the search path for resource files, such as the scan offset files and grayscale calibration curve files. If a resource file name is specified as a relative

name, the transform searches every directory in the path, in the order specified in the path, until it finds the file or fails. This search path does not apply to anything in the configuration file.

*directory* Specifies the search path for resource files. The default is **C:\Program Files\IBM Printing Systems\config**.

**-rot {0 | 90 | 180 | 270 [ p | i ]}**

Specifies how far to rotate the image and where the image is rotated.

The following values specify how far to rotate the image:

- 0** The image is not rotated.
- 90** The image is rotated clockwise 90 degrees.
- 180** The image is rotated clockwise 180 degrees.
- 270** The image is rotated clockwise 270 degrees.

The following values specify where the image is rotated:

- p** The printer rotates the image.
- i** The transform rotates the image. When **i** is specified, the transform takes longer to run and uses more memory.

**-scale {orig | a5 | a4 | a3 | letter | folio | legal | ledger | xxx% | xx.x,yy.y [ d | m | i | p ]}**

Specifies the output image size. The image is scaled to this size and centered in the area defined by the paper size and margin options.

**orig** Preserve the input image size, regardless of the printable paper area. The image still has to be scaled if the input file specifies non-square pels or is halftoned.

**a3** A3 format (297 by 420 mm or 11.69 by 16.54 inches)

**a4** A4 format (210 by 297 mm or 8.27 by 11.69 inches)

**a5** A5 format (148 by 210 mm or 5.83 by 8.27 inches)

**letter** 8.5 by 11 inches (216 by 279 mm)

**folio** 8.5 by 13.0 inches (216 by 330 mm)

**legal** 8.5 by 14.0 inches (216 by 356 mm)

**ledger** 11.0 by 17.0 inches (279 by 432 mm)

**xxx%** Scaling is relative to the original image size. Numbers below 100% mean reduction, while numbers above 100% mean enlargement. For example, **-scale 200%** causes the output image to be twice as long and twice as wide as the input image.

**xx.x,yy.y[d | m | i | p]**

Specifies the image dimensions.

**xx.xx** Specifies the horizontal dimension.

**yy.yy** Specifies the vertical dimension.

The following values specify the unit of measurement:

**d** Measurements are in dots (pels).

**i** Measurements are in inches.

**m**      Measurements are in millimeters.

**p**      Measurements are in points.

**-sgcorr** *FileName*

Specifies the name of the scanner-related calibration file. If **-calib** is not specified, this option is ignored.

*FileName*      The name and location of the scanner calibration file. If there are spaces in the file name and you are specifying this value on the command line, enclose the file name in double quotation marks.

**-sniff**

See [**-nosniff** | **-sniff**]

**-term**

See [**-noterm** | **-term**]

**-thresh** *FileName*

Specifies a file that contains a PostScript Type 1 or Type 3 halftone dictionary. Alternatively, the PostScript code can specify the **setscreen** operator instead of a Type 1 dictionary. The halftone cell in the dictionary is used to overwrite the default ordered dither clustered dot halftone cell. If the PostScript code contains a transfer function, either in the halftone dictionary or specified by the **settransfer** operator, the current grayscale correction curve is overwritten as if **-gcorr** were used.

*FileName*      The name and location of the file that contains an appropriate halftone dictionary. If there are spaces in the file name and you are specifying this value on the command line, enclose the file name in double quotation marks.

**-v**    See [**-nov** | **-v**]

**-w** {**8.5i** | *yyy.yy*[**d** | **m** | **i** | **p**]}

Specifies the paper width. See also **-l** to set the paper length or **-paper** to set both dimensions at the same time.

**8.5i**      The page width is 8.5 inches.

The following values specify the unit of measurement:

**d**      The width is measured in dots (pels).

**i**      The width is measured in inches.

**m**      The width is measured in millimeters.

**p**      The width is measured in points.

**-wrkdir** *WorkDirectoryName*

Input to **tiff2afp** from stdin must be cached before it can be processed. This option specifies which directory to use for caching the input. If this option is not specified, or the directory cannot be opened, the transform tries to use **C:\Program Files\IBM Printing Systems\tiff2afp** as the working directory. If the transform cannot use **C:\Program Files\IBM Printing Systems\tiff2afp**, it uses **C:\temp**.

The stdin cache file is not visible to the user because it is immediately unlinked after creation. Thus, the cache file is always deleted when **tiff2afp** finishes, even if it ends abnormally.

**-x {0 | *nnn.nn*[*dl m i l p*]}**

Specifies the left margin.

**0**            There is no left margin.

*nnn.nn*

The left margin is *nnn.nn* units.

The following values specify the unit of measurement:

**d**            The margin is measured in dots (pels).

**i**            The margin is measured in inches.

**m**            The margin is measured in millimeters.

**p**            The margin is measured in points.

**-y {0 | *nnn.nn*[*dl m i l p*]}**

Specifies the top margin.

**0**            There is no top margin.

*nnn.nn*

The top margin is *nnn.nn* units.

The following values specify the unit of measurement:

**d**            The margin is measured in dots (pels).

**i**            The margin is measured in inches.

**m**            The margin is measured in millimeters.

**p**            The margin is measured in points.

---

## Transforming TIFF Data to AFP

This section describes how to transform TIFF data to AFP and gives examples.

### Input to the TIFF to AFP Transform

Input can be a TIFF file submitted by standard input (stdin) or specified on the command line. If no input file is specified, stdin is assumed. The extension .tif, .tiff, .TIF, or .TIFF does not need to be given explicitly. **tiff2afp** first tries to open the file as specified and, if unsuccessful, tries to append the extensions .tif, .tiff, .TIF, and .TIFF in turn.

### Specifying the Output File

The output file name and location can be derived from the input file name and location, specified on **-o**. Alternatively, you can force output to be standard output (stdout - this is typically output to your screen) by specifying **-o -** or **-o stdout** on the command line or **o = -** or **o = stdout** in the configuration file. If you request standard output, no output file is generated. By default, the output file has the same name as the input file and is stored in the same location as the input file.

If the output file name is not specified explicitly, the transform strips any extension from the input file name and appends the appropriate extension to create the output file name.

For example, all the following commands have *myfile.afp* as the output file:



```
tiff2afp myfile
tiff2afp myfile.tiff
tiff2afp myfile -o myfile.afp
```

Note that there is no requirement for the explicitly specified input and output files to have .tiff and .afp extensions. To process TIFF file foo.bar into an AFP file named foo.bar2, use this command to invoke the transform:

```
tiff2afp foo.bar -o foo.bar2
```

## Invoking tiff2afp

To invoke **tiff2afp**, go to a DOS prompt and specify `tiff2afp file -option value...` *file* is the location and name of the TIFF file and *option value* is any parameter you want to specify on the command line. You do not have to specify any parameters on the command line. The parameter values are determined by this hierarchy (least significant first):

1. Internal defaults
2. Environment variables
3. Configuration file
4. Command-line arguments

### Notes:

1. If there are spaces in the path or name, enclose the string in double quotation marks. For example, "C:\Program Files\IBM Printing Systems\files\file.tiff".
2. If the directory that contains **tiff2afp.exe** is not in your PATH specification, you must be in that directory to run the command. If you accepted the default directories at install time, this directory is **C:\Program Files\IBM Printing Systems\bin**. See "Editing Your Windows Path" on page 77 for instructions to change your PATH.

## Command Line Arguments

When using the command line, note the following:

- Options and input file names can be specified in any order.
- Only file names are case-sensitive.
- If an option is specified multiple times, only the last one is recognized.
- If multiple configuration files are given, they are processed in the order in which you specify them.

## Examples

### Example 1

This example transforms a TIFF file called MyOverlay.tiff into an AFP overlay called MyOverlay.afp. The output file name is not specified because it is given the same name as the input file with the .afp extension by default. It is stored in the specified directory so it can easily be imported to the system MyIseries. Enter this command at a DOS prompt:

```
tiff2afp MyOverlay.tiff -pagetype overlay -o \\MyIseries\QDLS\
```

The same results can be achieved with this command and configuration file:

```
tiff2afp MyTiffFile.tiff -C ConfigFile1
```

ConfigFile1 contains this:

```
pagetype=overlay
o=\\MyIseries\QDLS\
```

## Example 2

This example transforms a TIFF file called `BigLogo.tiff` into an AFP page segment called `LOG0`. The logo needs to be scaled down and rotated 90 degrees. To have greater control over which pels are colored, we specify the `alg2` algorithm and specify that a pel in the output is only colored if 80% of the pel was colored in the input file. However, specifying `alg2` makes the transform take longer to complete. The output is stored in the specified directory so it can easily be imported to the system `MyIseries`. Enter this command in a single line at a DOS prompt:

```
tiff2afp BigLogo.tiff -pagetype pseg -scale 2,2i -rot 90 -alg alg2 -clean .8  
-o \\MyIseries\QDLS\LOG0.afp
```

The same results can be achieved with this command and configuration file:

```
tiff2afp BigLogo.tiff -C ConfigFile2
```

`ConfigFile2` contains this:

```
pagetype=pseg  
scale=2,2i  
alg=alg2  
clean=.8  
o=\\MyIseries\QDLS\LOG0.afp
```

---

## Error Recovery

The transform can encounter two types of errors; fatal and nonfatal.

When the transform finds a fatal error, the last output page might be incomplete. These are fatal errors:

- Errors in the user-specified parameters; either in the environment variables, the configuration file, or command line arguments. These errors include unrecognized options, incorrect values for parameters, or inability to open the configuration file or the output file.
- Out of memory errors. The requested memory is larger than the memory specified using the **-M** option or the dynamic memory allocation has failed.
- Disk space errors. If the file system is full, the transform deletes the partial output file before terminating. See the **-ms** and **-msf** options for setting the minimum free disk space requirements.
- Internal errors.

When the transform finds a nonfatal error in a file, the transform tries to recover with the next image in the file, and a warning message is issued. Errors and warnings are output to standard error.

A nonfatal error is made fatal if **-term** is specified. Nonfatal errors include errors that are encountered in the TIFF files:

- The transform cannot open the input file.
- The input file is not TIFF.
- The input file contains features that cannot be processed.

---

## Limitations

The **tiff2afp** transform has these limitations:

- Only Huffman coding with baseline DCT JPEG algorithm is supported.

- Old style JPEG specification (Photometric Interpretation 6) is not supported. Use the Photometric Interpretation 7 to incorporate the JPEG compressed images in TIFF files.
- Additional color information, such as alpha data, is disregarded.
- Transparency maps are ignored.
- At most 8-bit grayscale and 24-bit color is supported.

---

## Acknowledgments

This software is partially based on the Independent JPEG Group's JPEG compression and decompression code.



---

## Chapter 6. Using the PDF Subsystem

Using the PDF subsystem, Infoprint Server can transform a spooled file to PDF and send the PDF file as electronic mail (e-mail). Alternatively, it can store the PDF file in the integrated file system or put it on an output queue. The input spooled file can be any type of data that PSF for OS/400 can print.

**Note:** The PDF subsystem interacts with PSF for OS/400 (hereafter referred to as PSF) to process data. However, you do not need a license for PSF to use the PDF subsystem.

This chapter contains this information:

“Overview” explains how the subsystem works.

“Before Using the PDF Subsystem” on page 95 explains how to set up a PSF configuration object and configure the device.

“Transforming Data” on page 97 explains how to transform a file.

“Modifying the Input Data” on page 99 explains how to use DDS, the Infoprint Server Create AFP Data command, or AFP Toolbox to modify your input data.

“PSF Configuration Object Parameters” on page 100 lists the PSF configuration object parameters that are related to the PDF subsystem.

“Usage Notes” on page 106 explains what differences to expect between AFP and PDF output.

“Error Recovery” on page 110 describes where errors are sent. It also lists common problems and how to fix them.

---

### Overview

To transform a spooled file to PDF, PSF receives a spooled file and sends it to the PDF subsystem. The PDF subsystem transforms the spooled file and puts the PDF file in the specified location in the integrated file system or sends it back to PSF. PSF can deal with the PDF output in two ways. It spools the PDF file for printing or e-mails it. It can optionally use information from a user-defined PDF mapping program to customize the e-mail. Figure 15 on page 94 illustrates this process.

You can use a *user-defined PDF mapping program* (hereafter referred to as the PDF mapping program) to further customize the PDF subsystem. A PDF mapping program is a versatile user exit program. You can use it to interpret your mail tags, specify the e-mail subject, and add text to the beginning of each e-mail. If you use an SMTP server to send the e-mail, you can also use the PDF mapping program to specify carbon copy (cc), blind carbon copy (bcc), and return-to addresses, a file to use as the e-mail body, and attachments.

You do not need to use a PDF mapping program. You can accept default text for the e-mail subject and body and specify a valid e-mail address on the USRDFNDDTA field in the printer file or the spooled file, or in the data. You can use DDS keywords, the CRTAFPDITA command, or AFP Toolbox to specify the e-mail address in the data.

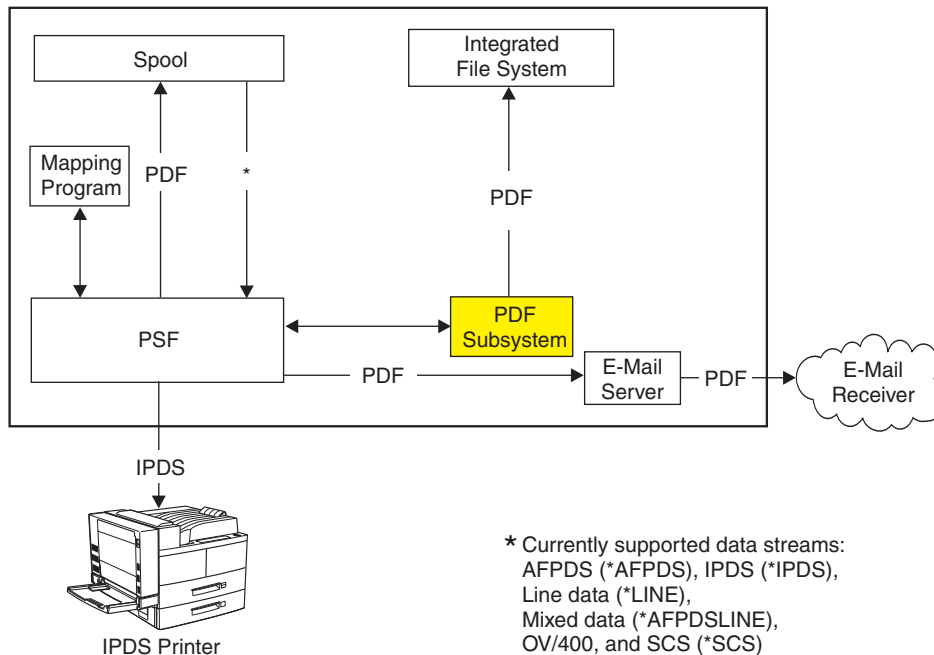


Figure 15. Transforming a Spooled File to PDF

If the input to the PDF subsystem is text-based, the input controls and data are converted into PDF files with text, graphics, and image data. The PDF files can be used in e-mail applications, or Internet-based softcopy repositories, or printed on PDF printers. The transform emulates a TCP/IP-connected IPDS 4028, 3812, or Infoprint 40 printer device and the conversion takes place at print time.

All resources (fonts are optional) are embedded in the PDF file, which guarantees resource availability. All embedded 240-pel fonts and AFP resources are converted to 300 pel during conversion. This transform supports single-byte fonts and downloaded double-byte raster fonts. Double-byte outline fonts and resident double-byte fonts are not supported.

#### Notes:

1. If you want to use your own IPDS to PDF transform instead of the PDF subsystem, you can use the PDF transform exit point. For information about using the PDF transform exit point, see Appendix D, "Exit Points" on page 163.
2. If the input to the PDF subsystem is an image file, the PDF output is image.
3. Any datastream transforms that occur before the input data is spooled might cause image data to be input to the PDF subsystem.

## Default Transform Characteristics

The transform output is controlled by the default transform characteristics and by the values specified in a PSF configuration object. The PDF subsystem accepts and transforms print jobs as if it was an actual IPDS printer with these basic characteristics:

- Two paper trays
- Duplex enabled
- Default output bin

If your print job uses other options, a message is sent and the document is transformed or is cancelled. This is controlled by the document fidelity setting specified in the printer file associated with the job.

**Note:** The resulting PDF output does not contain duplex or input bin settings.

The PDF subsystem emulates an IPDS printer with these IPDS configuration settings defined in the printer. The equivalent Infoprint 11xx settings have been included for those who are more familiar with Infoprint 11xx settings. The Infoprint 11xx menu items are found in the IPDS MENU menu.

*Table 3. Default IPDS Configuration Settings*

4028, 3812, or Infoprint 40 Setting	Infoprint 11xx Menu Item
vpa chk = off	EMULATION/Exception Control = Sup Beyond VPA, Suppress Both, or Sup Undef char
x offset = 0	MARGINS/(paper source)/Left Margin = 0
y offset = 0	MARGINS/(paper source)/Top Margin = 0
page = whole	EMULATION/Printable Area = 4028 Whole Page
edge-edge = on	EMULATION/Printable Area = Physical Page (edge-to-edge)
font sub = on	EMULATION/Font Ctrl = Relaxed
gcs = char	No equivalent value
scal bar code = auto	No equivalent value
box draw = off	No equivalent value

Use the PDFDEVTYPE option in the PSF configuration object to further specify IPDS printer characteristics.

---

## Before Using the PDF Subsystem

This section describes what you have to do before using the PDF subsystem. In order to use the subsystem, you need to follow the instructions in these sections:

1. “Setting up Your PSF Configuration Object”.
2. “Configuring the Device” on page 96.
3. “Varying the Device ON” on page 96.
4. “Starting the Writer” on page 97.

## Setting up Your PSF Configuration Object

Before you set up your PSF configuration object, you need to decide what you want done with the PDF file after it is created. It can be stored in a directory, placed on an output queue, or sent as e-mail.

Create a new PSF configuration object or change an existing PSF configuration object by using CRTPSF CFG or CHGPSF CFG, respectively. You must specify something other than \*NO for PDFGEN on the PSF configuration object in order to use the PDF subsystem. The parameters that let you customize a PSF configuration object for use with the PDF subsystem are listed in “PSF Configuration Object Parameters” on page 100.

Instead of using the command line, you can create and change a PSF configuration object by using the iSeries Navigator component of iSeries Access. Refer to *iSeries Guide to Advanced Function Presentation* for instructions.

For descriptions of other parameters you can use with CRTPSFCFG and CHGPSFCFG, refer to the appropriate command description topic in the iSeries Information Center.

**Example:**

This command creates a PSF configuration object that tells the PDF subsystem to generate PDF and send it as e-mail. It accepts default values for all other parameters.

```
CRTPSFCFG PSFCFG(name_of_configuration_object) PDFGEN(*MAIL)
```

## Configuring the Device

Set up your device the same way you would set up a TCP/IP IPDS printer. You can have multiple devices active, but each must be configured as a different printer with a different port. Use Create Device Description (Printer) (CRTDEVPRT) or Change Device Description (Printer) (CHGDEVPRT) to specify the following:

Device description	<i>device-name</i>
Device class	*LAN
Device type	*IPDS
Device model	0
LAN attachment	*IP
Advanced function printing	*YES
Port number	xxxxx, the number is unique for each printer.
Font	<i>printer-font</i>
Remote location	127.0.0.1
User defined object	
	<b>Object</b> <i>PSF-configuration-object-name</i>
	<b>Library</b> <i>PSF-configuration-object-library</i>
	<b>Object type</b> *PSFCFG

**Example:** In this example, we configure the device named PRINTER. It is set up on port number 12345, and jobs sent to this device use font 001 by default. The PSF configuration object associated with PRINTER is MYPSFCFG. “Setting up Your PSF Configuration Object” on page 95 describes how to create an appropriate PSF configuration object.

```
CRTDEVPRT DEVD(PRINTER) DEVCLS(*LAN) TYPE(*IPDS)
MODEL(0) LANATTACH(*IP) AFP(*YES) PORT(12345) FONT(001)
RMTLOCNAME('127.0.0.1') USRDFNOBJ(MYLIB/MYPSFCFG *PSFCFG)
```

## Varying the Device ON

Enter this command to vary the device ON:

```
VRRCFG CFGOBJ(device-name) CFGTYPE(*DEV) STATUS(ON)
```

*device-name* is the name of the device created in “Configuring the Device”.



**Example:** In this example, we vary the device named PRINTER on.  
VRYCFG CFGOBJ(PRINTER) CFGTYPE(\*DEV) STATUS(ON)

## Starting the Writer

Enter this command to start the printer writer:  
STRPRTWTR(*device-name*)

**Example:** In this example, we start the writer named PRINTER.  
STRPRTWTR(PRINTER)

---

## Transforming Data

There are two ways to access the PDF subsystem:

- Submit the job to a printer that is set up properly. See “Configuring the Device” on page 96 for information about setting up the printer device. See “Example” for an example.
- Use the iSeries Navigator component of iSeries Access. See “Accessing the PDF Subsystem with iSeries Access” on page 98 for instructions.

You use the PSF configuration object associated with the printer to specify what is done with the PDF file and other transform options. For information about PSF configuration object parameters you can use to customize the PDF transform, see “PSF Configuration Object Parameters” on page 100.

## Example

This example shows how to convert the file MY\_FILE to PDF and send it as e-mail.

1. Set up the OS/400 as a POP server. This lets us use our local OS/400 as an SMTP server to send the e-mail. For instructions, refer to “Setting up POP e-mail clients” in the iSeries Information Center.
2. Create the user profile ME and give it an SMTP user ID and SMTP domain.
  - This command creates a user ID called ME in the System Distribution Directory. The user’s address is ADDRESS and the user works on the system SYSTEM. The description is “My user profile”.  
ADDIRE USRID(ME ADDRESS) SYSNAME(SYSTEM) USRD(My user profile)
  - Because we are using an SMTP mail server to send e-mail, ME needs to have an SMTP address. Follow these steps to add the SMTP address ME@BUSINESS.COM:
    - a. Enter this command: WRKDIRE USRID(ME ADDRESS)
    - b. Specify option 2 “Change” in the Opt field next to ME.
    - c. Press F19, “Change name for SMTP”.
    - d. Press Enter when prompted at the bottom of the screen.
    - e. Enter the SMTP User ID (ME) and Domain (BUSINESS.COM) on the appropriate lines.
3. Create the PSF configuration object. In this example, the PSF configuration object is called EMAILS. It specifies that the local OS/400 is used to send the e-mail and that the user ME is the e-mail sender.  
CRTPSFCFG PSFCFG(EMAILS) PDFGEN(\*MAIL) PDFMAILSVR(\*LOCAL) PDFSENDER(ME)
4. Specify the e-mail address. This e-mail is to be sent to JOE@IBM.COM. There are many ways to specify the e-mail address. For more information about

specifying an e-mail address, see Chapter 8, “Sending E-mail” on page 129. In this example, we put the e-mail address in the printer file MY\_PRTF, which is associated with the job.

```
CHGPRTF FILE(MY_PRTF) USRDFNDA('MAILTAG(JOE@IBM.COM)MAILSENDER(ME)')
```

5. Configure the device. This command creates a new device description called MAIL. Note that EMAILS is the PSF configuration object we created to convert an input file to PDF and e-mail it.

```
CRTDEVPRT DEVD(MAIL) DEVCLS(*LAN) TYPE(*IPDS) MODEL(0) LANATTACH(*IP) AFP(*YES)  
PORT(12345) RMTLOCNAME(127.0.0.1) USRDFNOBJ(EMAILS *PSFCFG)
```

6. Vary the device ON:

```
VRYCFG CFGOBJ(MAIL) CFGTYPE(*DEV) STATUS(ON)
```

7. Start the writer:

```
STRPRTWTR(MAIL)
```

8. Send the spooled file MY\_FILE to the writer:

```
PRTAFPDTA FILE(MY_FILE) DEV(MAIL)
```

## Accessing the PDF Subsystem with iSeries Access

You can use iSeries Navigator to access the PDF subsystem. To do this, you need AFP Manager and Infoprint Server for iSeries 5.2 installed. *iSeries Guide to Advanced Function Presentation* has instructions for installing AFP Manager.

### Before You Begin

- Make sure that you have a PSF configuration object that tells PSF/400 to transform the spooled file into a PDF file.  
Use one of these methods to create a new PSF configuration object or modify an existing one:
  - Use AFP Manager. Refer to *iSeries Guide to Advanced Function Presentation* for instructions. On the PDF Transform page, select **Generate PDF**. For help on any field, select the field and press F1.
  - Run the command Create PSF Configuration (CRTPSFCFG) or Change PSF Configuration (CHGPSFCFG) from iSeries Navigator. Specify a value other than \*NO for Generate PDF (PDFGEN). You can specify other appropriate values as described in “Setting up Your PSF Configuration Object” on page 95. Refer to *iSeries Guide to Advanced Function Presentation* for instructions to run an OS/400 command from iSeries Navigator.
  - Run the command Create PSF Configuration (CRTPSFCFG) or Change PSF Configuration (CHGPSFCFG) from the OS/400 command line. Specify a value other than \*NO for Generate PDF (PDFGEN). You can specify other appropriate values as described in “Setting up Your PSF Configuration Object” on page 95.
- Make sure that the writer that processes the spooled files is configured properly.  
To configure a writer to transform a spooled file to PDF, run the command Create Device Description, Printer (CHGDEVPRT) or Change Device Description, Printer (CHGDEVPRT). Specify the values listed in “Configuring the Device” on page 96. You can run commands from iSeries Navigator or the OS/400 command line. Refer to *iSeries Guide to Advanced Function Presentation* for instructions to run an OS/400 command from iSeries Navigator.

**Note:** Do not change the User data field on the input spooled file.

## Steps to Convert a Spooled File to PDF

1. In iSeries Navigator, expand **My Connections** (or your active environment).
2. Expand **Basic Operations** and select **Printer Output**.
3. Right-click the output file you want to convert and select **Convert to PDF...** . To select more than one file, CTRL+click each file. The **Convert Printer Output to PDF** dialog opens with the specified printer output files in the **Printer output list**.
4. Select **Send as electronic mail**, **Save in Integrated File System**, or **Save to output queue** along with any other values you want, such as a PDF mapping program.
5. Specify the device to use to convert the printer output file. You can specify library-qualified output queue, the printer to send the output file to, or both.
6. Click **OK**.

A new printer output file is created for each printer output file you selected. When the file is processed, the new output files are removed from the output queue. For help with any field, select the field and press **F1**.

**Note:** Do not change the User data field on the input spooled file.

---

## Modifying the Input Data

This section describes how to use different tools to modify your input data for use with the PDF subsystem. It describes how to use DDS keywords, the Infoprint Server Create AFP Data command, and AFP Toolbox. Modifying the input data is optional.

You can insert group tags into your data before it is transformed. This lets the subsystem generate one PDF file for each group or insert an index tag at the beginning of each group. You can also use group tags to e-mail each group to a different recipient. The group name is used as the index tag or mail tag when appropriate. For information about putting mail tags on the group tags, see "Inserting a Mail Tag in the Input File" on page 134.

You can use a tool such as DDS, AFP Toolbox, or the Create AFP Data (CRTAFPDTA) command to insert group tags in your data. The value specified for PDFMULT on the PSF configuration object determines how the PDF subsystem uses the group tags. It either generates one PDF file for every group or places one index tag at the beginning of each group. If you choose to specify groups, all of the data must be inside a group.

## Using DDS to Modify the Input File

For DDS print applications, you can put a mail tag on the DDS STRPAGGRP keyword. The STRPAGGRP keyword lets you specify up to 250 characters for the mail tag. This can be an actual email address or it can be a keyword, such as a customer name. A keyword is mapped to one or more email addresses by a PDF mapping program (PDFMAPPGM parameter on PSF configuration object). For each STRPAGGRP keyword, there must be a corresponding ENDPAGGRP keyword. The DOCIDXTAG keyword is not used for mail tags.

### Notes:

1. The ENDPAGGRP keyword must be issued before ending the page (with the ENDPAGE keyword, for example).
2. DEVTYPE(\*AFPDS) must be specified on the printer file.

3. If you use any STRPAGGRP and ENDPAGGRP keywords, all data must be inside a page group.

For more information about using DDS, refer to *DDS Reference*.

## Using AFP Toolbox to Modify the Input File

To use AFP Toolbox to generate groups, use the Begin Group program call to create Begin Named Group (BNG) structured fields. Refer to *AFP Toolbox for Multiple Operating Systems: User's Guide* for more information about using AFP Toolbox.

## Using CRTAFPDTA to Modify the Input File

To use CRTAFPDTA to generate groups, follow the instructions in "Indexing with CRTAFPDTA" on page 12 to index your file. The **Index tag definition** (IDXTAG) value that is specified with **Select index for group names** (IDXGRP) is the group name. For example, in "Indexing with CRTAFPDTA" on page 26, the default value for IDXGRP, \*IDXTAG1, is NAME. This means that the value in the first index tag, which is labelled NAME, is used as the mail tag. Because the values in the index tag are not valid e-mail addresses, you would need a PDF mapping program to map those values to actual e-mail addresses. See Appendix B, "PDF Mapping Program" on page 143 and "PDF Mapping Program in C" on page 155 for information about using a PDF mapping program.

---

## PSF Configuration Object Parameters

You can use these parameters to customize the PDF transform part of the PDF subsystem. You must specify something other than \*NONE for PDFGEN in order to transform data to PDF.

### PDFDEVTYPE

To transform a spooled file to PDF, you send the file to the PDF subsystem's virtual printer device. This parameter specifies the type of device that the PDF subsystem's virtual printer device emulates. This parameter is only valid if PDFGEN has a value other than \*NONE. Options are:

#### \*IP40240

The virtual printer device emulates an Infoprint 40 printer device at 240 dots per inch (dpi) resolution. The transform works like a TCP/IP connected 240 ppi IPDS Infoprint 40 with the 4820 feature, with these exceptions:

- Object area shading and page area shading is supported.
- Bi-level color image and text, and graphic and bar code color is preserved in PDF output.
- Multiple output bins, finishing, and jogging are not supported.
- IPDS multi-resolution support is not supported.
- Multiple identical copies are not supported.
- Separator pages are not supported.
- Double-byte outline fonts are not supported. There are no resident double-byte fonts.

**\*IP40240** can use raster fonts and outline fonts raster and outline fonts.

#### \*IP40300

The virtual printer device emulates an Infoprint 40 printer device

at 300 dpi resolution. The transform works like a TCP/IP connected 300 ppi IPDS InfoPrint 40 with the 4820 feature, with these exceptions:

- Object area shading and page area shading is supported.
- Bi-level color image and text, and graphic and bar code color is preserved in PDF output.
- Multiple output bins, finishing, and jogging are not supported.
- IPDS multi-resolution support is not supported.
- Multiple identical copies are not supported.
- Separator pages are not supported.
- Double-byte outline fonts are not supported. There are no resident double-byte fonts.

**\*IP40300** can use raster and outline fonts. IBM recommends that you use this option whenever possible.

- \*4028** The virtual printer device emulates a 4028 printer device. This value is designed to be used for compatibility only and uses the raster fonts that are resident in a 4028 printer. It cannot use outline fonts.
- \*3812** The virtual printer device emulates a 3812 printer device. This value is designed to be used for compatibility only and uses the raster fonts that are resident in a 3812 printer. It cannot use outline fonts.

## PDFDIR

Specifies the directory where the stream file will be stored. The directory must be created by the user or installation. This parameter is required if PDFGEN(\*STMF) is specified. If you want to mail this file later using the Send Distribution (SNDDST) command, specify QDLS/directory-name as the directory name.

The spool writer (QSPLJOB) needs at least execute (\*X) authority to every directory in the path and also needs change (\*RWX) authority to the directory that will contain the PDF files (the parent directory). For example, if you want the PDF files put in directory3, which is in this path:  
/directory1/directory2/directory3, QSPLJOB needs \*X authority to directory1 and directory2 and \*RWX authority to directory3.

You can grant authority to the writer in one of two ways. You can either specify the appropriate authority for Public, or you can grant appropriate private authority to the QSPLJOB user profile.

The additional subdirectories that the transform needs are created by the PDF subsystem. The new PDF file is owned by the spooled file owner and has public authority of \*EXCLUDE. This parameter is required and is only valid if PDFGEN(\*STMF) is specified. Options are:

### *PDF-directory-name*

Specifies the directory where the stream file is stored.

If you specify a QDLS directory such as folder-name, the file will be stored in this location:

/QDLS/folder-name/job-name/job-number/job-user-name/file-number  
/spooled-file-name/date/sequence-number/

The path has these values:

**job-name**

The job name assigned to the spooled file. There is an 8-character segment name limitation on QDLS path names. Therefore, the job name is left-aligned and the last two characters are stored as a prefix on the job number.

**job-number**

The job number assigned to the spooled file, prefixed by the last two characters of the job name.

**job-user-name**

The job user name assigned to the spooled file. There is an 8-character segment name limitation on QDLS path names. Therefore, the job user name is left-aligned and the last two characters are stored as a prefix on the file number.

**file-number**

The file number assigned to the spooled file, prefixed by the last two characters of the job user name.

**date** The two-digit month appended to the two-digit day appended to the four-digit year.

**spooled-file-name**

The first eight characters of the spooled file name.

**sequence-number**

The six character sequence number. It is 000001 if PDFMULT(\*NO) is specified. If PDFMULT(\*YES) is specified, the sequence number is incremented to uniquely identify each PDF file generated for a job, starting with 000001. If an error is encountered in the conversion, the sequence number of the file is prefixed with an E.

If you specify a Root File System directory name such as PDF-directory-name, the file will be stored in this location:

```
/PDF-directory-name/job-name/job-user-name
/job-number_file-number_spooled-file-name_date_sequence-number
```

The path has these values:

**job-name**

The 10-character job name assigned to the spooled file.

**job-user-name**

The 10-character user name.

**job-number\_file-number\_spooled-file-name\_date\_sequence-number**

The six-character job number, appended to the six-character file number, appended to the 10-character spooled file name, appended to the eight-character date (month-day-year), appended to a six-character sequence number (see **sequence-number** above). Each value is separated by an underscore.

**PDFDTAQ**

Specifies the name and library of the data queue where PDF logs the notifications that a transformation has been completed. This parameter is only valid if PDFGEN has a value other than \*NONE. Keyed data queues are not

supported. For information about using this data queue, see “PDF Conversion Completion Data Queue” on page 112. Options are:

**\*NONE** No completion notifications are logged in a data queue.

*Library-name/PDF-data-queue*

Specifies the name of the data queue and the library it is in. PSF does not create or manage this data queue.

## PDFGEN

Specifies whether the spooled file is converted to PDF. This parameter is valid only for printer devices that are configured with the loopback address and AFP(\*YES). Options are:

**\*NONE** No PDF file is generated.

**\*SPLF** The PDF output is placed on the output queue specified in PDFOUTQ. You must also specify a value for PDFOUTQ.

**\*STMF** The PDF output is placed into a stream file in the directory specified in the PDFDIR parameter. You must also specify a value for PDFDIR.

**\*MAIL** The PDF output is electronically mailed and then erased. You can specify a value for PDFSENDER or accept the default. The PDF file is sent as an attachment. If your OS/400 is not set up to send e-mail, see “Enabling Your OS/400 to Send E-Mail” on page 130 for instructions.

For sending e-mail, you can also use these parameters: PDFMULT, PDFSENDER, and PDFMAPPGM. For more information about sending a PDF file as e-mail, see Chapter 8, “Sending E-mail” on page 129.

## PDFINCFNT

Specifies whether the PDF files generated by the PDF subsystem carry the necessary fonts inline.

**\*YES**

Fonts are carried inline with the PDF output. This makes the file larger, but guarantees font fidelity.

**\*NO**

Fonts are not carried inline with the PDF output. The application needs access to the fonts for printing or viewing the file.

## PDFMAILSVR

Specifies which mail server PSF uses to e-mail the PDF output files. You can use an SMTP server or the Send Distribution (SNDDST) command. This parameter is only valid if PDFGEN(\*MAIL) is specified. If you specify an OS/400 to be your SMTP server, it must be configured as a POP server. For instructions, refer to “Setting up POP e-mail clients” in the iSeries Information Center.

Options are:

**\*SNDDST**

The Send Distribution (SNDDST) command is used to send e-mail.

**\*LOCAL**

Your local OS/400 is used as an SMTP server. It must be configured as a POP server.



#### *mail-server-name*

Specify up to four SMTP mail servers to try to use to send e-mail. \*LOCAL can be one of the four. To use an OS/400, it must be configured as a POP server.

PSF tries to use the first server listed to send the e-mail. If the first server does not work, PSF tries each remaining server in order until one works. If none of the mail servers can be used, the printer is ended or an inquiry message is issued. The action taken depends on the value specified for the Print Error Message (PRTERRRMSG) parameter of the printer's device description.

#### **PDFMAPPGM**

Specifies the library-qualified name of a user-defined PDF mapping program. You can use a PDF mapping program to interpret your mail tags, specify the e-mail subject, and add text to the beginning of each e-mail. If you use an SMTP server to send the e-mail, you can also use the PDF mapping program to specify carbon copy (cc), blind carbon copy (bcc), and return-to addresses, a file to use as the e-mail body, and attachments.

If a PDF mapping program is not specified, PSF assumes that the mail tags are valid e-mail addresses (in the form *name@domain*). It will try to use the information in the mail tag to send the file.

If a PDF mapping program is specified, PSF assumes that all mail tags need to be mapped. It must exist when the PSF configuration object is created.

For more information about using the PDF mapping program, see Appendix B, "PDF Mapping Program" on page 143. For information about the required structure of the PDF mapping program, see Appendix C, "Templates" on page 155.

This parameter is only valid if PDFGEN(\*MAIL) is specified. Options are:

##### **\*NONE**

There is no user program. PSF assumes that the mail tag is a valid e-mail address and tries to use the information in the mail tag to send the file.

#### *Library-name/PDF-mapping-program-name*

Specifies the name of the PDF mapping program to use and the library it is in.

#### **PDFMULT**

Specifies whether the transform splits the PDF output into multiple files at group boundaries. This parameter is only valid if PDFGEN has a value other than \*NONE. All of the PDF files are processed the same way. For example, you can subdivide a customer statement run on customer statement boundaries and e-mail each customer the appropriate statement. You cannot store one PDF file and e-mail the rest. For information about creating an input file that generates multiple output files, see "Modifying the Input Data" on page 99. Options are:

##### **\*NO**

One PDF file is created

##### **\*YES**

Multiple PDF files are processed when applicable.

##### **\*SPLIT**

Multiple PDF output files are generated.

##### **\*INDEX**

An index tag is placed at the group boundaries but only one output file is generated. If you want to view the PDF file, this option is useful for navigation.



## PDFOUTQ

Specifies the library-qualified name of an output queue that PSF uses when spooling the PDF output. You will get error messages if the output queue does not exist or you do not have authority to the queue at run time. This parameter is required when PDFGEN(\*SPLF) is specified. The owner of the original spooled file becomes the owner of the new PDF file. PDFOUTQ requires this value:

*Library-name/PDF-output-queue*

Specifies the name of the output queue to use and the library it is in.

## PDFPPRDWR1

Specifies the paper size to use for drawer one during the transform process. This information is used as the page size in the PDF output file. The paper size is reported to PSF for the IPDS printer device and is considered the printable area.

This parameter is only valid if PDFGEN has a value other than \*NONE. Options are:

### **\*LETTER**

Use the dimensions of letter paper.

### **\*LEGAL**

Use the dimensions of legal paper.

### **\*STATEMENT**

Use the dimensions of statement paper.

### **\*EXECUTIVE**

Use the dimensions of executive paper.

### **\*LEDGER**

Use the dimensions of ledger paper.

**\*A5** Use the dimensions of A5 paper.

**\*A4** Use the dimensions of A4 paper.

**\*A3** Use the dimensions of A3 paper.

**\*B5** Use the dimensions of B5 paper.

**\*B4** Use the dimensions of B4 paper.

## PDFPPRDWR2

Specifies the paper size to use for drawer two during the transform process. This information is used as the page size in the PDF output file. The paper size is reported to PSF for the IPDS printer device and is considered the printable area.

This parameter is only valid if PDFGEN has a value other than \*NONE. Options are:

### **\*LETTER**

Use the dimensions of letter paper.

### **\*LEGAL**

Use the dimensions of legal paper.

### **\*STATEMENT**

Use the dimensions of statement paper.

### **\*EXECUTIVE**

Use the dimensions of executive paper.

**\*LEDGER**

Use the dimensions of ledger paper.

**\*A5** Use the dimensions of A5 paper.

**\*A4** Use the dimensions of A4 paper.

**\*A3** Use the dimensions of A3 paper.

**\*B5** Use the dimensions of B5 paper.

**\*B4** Use the dimensions of B4 paper.

**PDFSENDER**

Specifies the sender from whom the PDF output file is e-mailed. This parameter is only valid if PDFGEN(\*MAIL) is specified. The sender must be a valid user profile and be enrolled in the System Distribution Directory.

If you specify PDFMAILSVR(*mail-server-name*) or PDFMAILSVR(\*LOCAL), the entry in the System Distribution Directory must have an SMTP user ID and SMTP domain specified. For example, if the sender's e-mail address is *name@business.com*, the SMTP user ID is *name* and the SMTP domain is *business.com*. For instructions about specifying an SMTP address on a System Distribution Directory entry, see step 7 in "Steps to Enable Your OS/400 to Send E-mail" on page 130.

Use Display Directory Entries (DSPDIRE) to view the entry in the System Distribution Directory and Work with Directory Entries (WRKDIRE) to change an entry. For more detailed information about working with the System Distribution Directory, see "Enabling Your OS/400 to Send E-Mail" on page 130.

The spooled file owner is responsible for any e-mail cleanup. Options are:

**\*SPLFOWN** The owner of the original spooled file sends the e-mail.

**QSPLJOB** QSPLJOB sends the e-mail.

*Mail-file-sender*

Specifies the name or a valid user profile that is the e-mail sender.

---

## Usage Notes

There are many considerations that affect the transform and how you can use the transformed data. This section describes restrictions on the AFP input and the differences to expect between the AFP and PDF files. It also describes how to best use the PDF output. See these sections for more information:

- "Bar Codes" on page 107
- "Color" on page 107
- "Error Checking" on page 107
- "Find Function" on page 107
- "Fonts" on page 107
- "Other Unsupported IPDS Functions" on page 108
- "Output File Size" on page 108
- "Page Numbering" on page 109
- "Printing Considerations" on page 109
- "Use of Printer Features" on page 109
- "Viewing Considerations" on page 110

## Bar Codes

The PDF transform does not support bar code types X'1A': RM4SCC, X'1B': Japan Postal Barcode, or 2D bar codes, including PDF 417, Data Matrix, and Maxicode. Bar codes generated by the PDF subsystem are not guaranteed to scan correctly.

## Color

The DDS color support that was available before OS/400 V5R1 can be used with the PDF output. For DDS extended color support available with OS/400 V5R1 and higher, the RGB and CYMK color models are supported. The CIELAB and HIGHLIGHT color models are not supported.

## Error Checking

The data stream error reporting in the PDF transform is not an exact emulation of the error checking by an IPDS printer. It is recommended that you print the PDF output to an actual IPDS printer before sending it as e-mail.

## Find Function

The Adobe Acrobat find function is designed to locate text strings within a document. Under certain circumstances, the find function might not work with transformed PDF files.

### **AFP character encoding representation**

The PDF transform uses the AFP font encoding to create the PDF document. The characters contained in the document might not be the same as the ASCII character mapping available on the keyboard used for PDF viewing.

### **Character alignment**

AFP text placement can use relative and absolute placement of characters within a single print line. In order to preserve the output fidelity of a document, corresponding placement of characters is performed in the PDF output document. This can result in additional "space" characters within a character string. This restricts the operation of the find function.

For best results with the Find function, search for single words.

### **Graphic characters**

The Find function cannot find text that has been generated using AFP GOCA output graphic characters.

## Fonts

AFP and IPDS output uses fonts that are resident on the OS/400 or are printer-resident. These fonts can be outline or raster in 240 or 300 dpi, depending on the printer emulation.

### **Font substitution**

When your document uses host-resident (downloadable) raster fonts, the PDF subsystem substitutes Adobe Type 3 fonts. When your document uses outline fonts or printer-resident raster fonts, the PDF subsystem substitutes Adobe Type 1 fonts. The PDF subsystem also uses Adobe Type 1 fonts if your document uses downloadable raster fonts and specify USEOUTLFNT(\*YES) in the PSF configuration object.

Because the Adobe Type 3 fonts appear blurry when viewing, IBM recommends you use outline or printer-resident fonts in your documents.

**Embedding**

When the IPDS document is converted to PDF, the font information is converted into PDF format and is embedded in the output document by default. This ensures that correct character encoding is retained and increases print fidelity, but increases file size.

If you specify PDFINCFNT(\*N0) on the PSF configuration object, fonts are not embedded. You should ensure that your viewing application has access to the correct fonts before using this option.

**Raster fonts**

Only the characters required for printing are embedded in the output document. This is called *font sieving*.

**Outline fonts**

IBM recommends that you use AFP outline fonts when possible for viewing and printing quality, and portability.

**Double-byte fonts**

The PDF transform does not support any resident double-byte character sets or downloaded outline double-byte character sets. Downloaded raster double-byte coded fonts are supported.

**Fixed pitch and PSM fonts**

Some of the fixed pitch and proportional spacing machine (PSM) fonts with FGIDs below 300 are not supported in IP40240 and IP40300 mode. Applications that require these fonts might have to use 4028 or 3812 mode.

**Default font and code page**

The default font and code page are specified by the CRTDEVPRT CHRID parameter.

## Other Unsupported IPDS Functions

The following IPDS functions are not supported by the PDF subsystem:

**Media controls:**

These media controls are not supported:

- Alternate media destination
- Media source by copy
- N\_Up printing

**Mark form processing:**

The size of the Mark Form will not always be the same as in the emulated IBM printer. No relation to the actual feed direction is implemented in the Mark Form indication.

**IPDS text controls**

Partial characters are not used for the Overstrike (OVS) operation.

## Output File Size

**Maximum size**

The PDF output cannot be larger than 10 GB.

**Outline fonts**

Each outline font referenced in the input document increases the PDF file size by approximately 110 KB.

## Page Numbering

PDF output contains a unique page number identification that is assigned during the creation of the PDF output file. This page number might not correspond to the page numbers used in the AFP input document.

The PDF output file cannot have more than 20,000 pages.

## Printing Considerations

The results of printed output may vary from the PDF document viewed using Adobe Acrobat. This is usually caused by the printer driver used.

### Adobe Acrobat

To print the PDF output from your desktop, you need Adobe Acrobat Reader 5.0 or higher. You can download it free from the Adobe Website: <http://www.adobe.com/products/acrobat/>.

### Printing from the spool

To print the PDF output on a PDF printer, that printer must support PostScript level 3015.

### Print fidelity

The quality and appearance of your PDF output depend on the type and capabilities of the printer you use.

### Printer specific features

If your input job is designed to use specific printer features, ensure that the device that runs the transform and the transform itself supports these capabilities. The printer features that the transform does not support are described in "Use of Printer Features".

### Edge-to-edge

Print output designed for edge-to-edge printing can only be accurately reproduced by a printer that supports edge-to-edge printing.

### Printable area

The transform specifies the printable area as the entire page size.

### Image output

PDF image output might look different than the original AFP image when the PDF document is printed, depending on your printer.

## Use of Printer Features

If your original document or formatting options requires specific media options or printer features, this information is not contained in the PDF output. For example, if a form definition or DDS specifies duplex printing, input bins, and output bins, this information is not contained in the PDF output. You can use printer driver options to select some of these options when you print the document.

### Page size

The page size used for PDF output depends on which paper drawer the input uses. The page size used by that paper drawer is specified on the PDFPPRDWR1 or PDFPPRDWR2 option in the PSF configuration object. The printable area is specified as the entire page size. The same page size is used for the whole PDF document.

### Duplex

Duplex information is not contained within the PDF document output. Thus,

documents that contain a combination of simplex and duplex pages will not retain that information. Use printer driver options to specify duplex when you print a PDF document.

#### **Number of copies**

PDF output is always produced as a single copy, regardless of DDS or form definition specifications. Use printer driver options to specify multiple copies when you print a PDF document.

#### **Landscape**

If you request landscape output, the PDF output will be in landscape format. If you later want portrait orientation, use Adobe Acrobat or the Acrobat Viewer Plug-in to rotate the document.

## **Viewing Considerations**

In certain circumstances, the PDF document might look different than the original AFP document.

#### **Adobe Acrobat**

The PDF output generated by the PDF transform is designed to be viewed with Adobe Acrobat Reader 3.0 or later, or Adobe Acrobat Plug-in 4.0. Use with earlier versions of these products might produce unpredictable results.

#### **Raster fonts**

The actual appearance of raster fonts when using Adobe Acrobat Viewer or Acrobat Plug-in can differ from the printed output. For example, in the Acrobat Reader some characters might not appear to be aligned on the character baseline. The appearance might change as you choose higher magnification levels in Acrobat Reader.

The default setting of the Acrobat Reader is to show all fonts under 6 pixels as Greek, or shaded gray lines. Therefore, some of the AFP output might not be visible in Acrobat Viewer. In order to correct this problem, in the Reader click **File** and select **Preferences**. On the **General** page, ensure that **Use Greek text below xx pixels** is not selected.

#### **Image output**

PDF image output might look different than the original AFP image when the PDF document is viewed, depending on your monitor.

---

## **Error Recovery**

This section describes where error messages are sent, common problems and possible solutions.

If the PDF subsystem encounters an error when starting the transform, the transform returns to the program that called it and writes an error message to the PDF transform job log. This will happen if TCP/IP is not started, for example. PSF times out when it tries to communicate with the PDF transform and logs its own error.

If an error occurs while processing an e-mail, the spooled file is held and processing continues with the next ready spooled file. This happens if the e-mail address is not in the form *name@domain* or has spaces on the end, for example.

## **Messages**

All error messages except 50C8 and 50CB are sent to the appropriate job log and printer message queue as they occur. Messages 50C8 and 50CB are only sent to

the job log. BATCH is the PDF subsystem job log. You can access the job logs through **job\_number/user/job\_name**. The printer message queue is specified on the CRTDEVPRT MSGQ parameter.

If you try to e-mail the PDF file to an OS/400 and the send fails, a failure message is issued to the message queue associated with the writer. To get this message when Send Distribution (SNDDST) sends the e-mail, the sender must use the Query Distribution (QRYDST) command with \*OUT specified on the Incoming or outgoing prompt (OPTION parameter).

If the e-mail is sent, but is not delivered, the non-delivery message is sent to the e-mail address specified as the PDFSENDER on the PSF configuration object. If the iSeries is configured as a POP server, you can use a mail program to retrieve the messages from the iSeries. For instructions about obtaining these messages when using SNDDST to send the e-mail, see “Checking for Non-delivery Messages” on page 136. For instructions about setting up your iSeries as a POP server, refer to the topic “Setting up POP e-mail clients” in the iSeries Information Center.

## Troubleshooting

This section describes some common errors and possible ways fix them.

Table 4. Troubleshooting the PDF Subsystem

Problem	Possible Solutions
The document does not print.	Examine the job log for error messages.  Change the print fidelity to content fidelity in the printer file.  Ensure you are using Adobe Acrobat Reader 5.0 or a printer that supports PostScript level 3015.
When I view the output, the fonts are blurry	Use outline or printer-resident raster fonts in your documents.  If your document uses host-resident (downloadable) raster fonts, specify USEOUTLFNT(*YES) in the PSF configuration object.  See on page 107 for more information.
The fonts look different in the PDF file.	Some font differences are normal. See “Printing Considerations” on page 109 or “Viewing Considerations” on page 110.  Make sure that the default settings have not been changed. See “Default Transform Characteristics” on page 94.  Use PDFDEVTYPE to select a different printer to emulate if font selection is not satisfactory.
Data is missing from the printout.	If the document is meant to print edge-to-edge, ensure that the printer is capable of printing edge-to-edge.  Use the <b>shrink to fit</b> option when printing.
A landscape page is presented in portrait orientation in Acrobat Viewer.	Use CTL+Shift+'+' to rotate the document clockwise 90 degrees in Acrobat Viewer version 4.0 or higher.

Table 4. Troubleshooting the PDF Subsystem (continued)

Problem	Possible Solutions
The printed output looks different from the original output.	Some differences are normal. See "Printing Considerations" on page 109.  If output is not acceptable, use a different printer or printer driver.
The information is not rotated on the page correctly.	The PDF subsystem rotates pages the same way a printer does. Use the PAGESIZE and PAGRTT parameters in the printer file to specify rotation. Refer to <i>Printer Device Programming</i> for information about rotating output.
The information is presented on the wrong size of paper.	Use the PDF paper size drawer parameters (PDFPPRDWR1 and PDFPPRDWR2) in the PSF configuration object to specify the page size.
When I specify that the PDF output is to be stored in the integrated file system and transform the job, the writer ends and there is a message indicating "security violation" in the job log.	Make sure your file permissions are correct. See on page 101 for more information.
When I specify that the PDF output is to be spooled to an output queue and transform the job, the writer ends and there is a message indicating "security violation" in the job log.	When the PDF output is spooled, it is stored in this temporary directory: QDLS/PS/writer-job-number/job-name/job-number/job-user-name/file-number/spooled-file-name/mmdyyy  QSPLJOB needs change (*RWX) authority to the PS directory. Any directories that do not already exist will be created with the appropriate authorities.

## PDF Conversion Completion Data Queue

In order to help customers write PDF mapping programs, Infoprint Server provides a data queue for tracking when a PDF conversion is complete. Entries are logged in this data queue when a spooled file has been converted to PDF and stored in the integrated file system or document library system. A PDF mapping program can use the Receive Data Queue API (QRCVDTAQ) to determine when a spooled file has been converted to PDF. Refer to *System API Programming* for more information about data queues and the QRCVDTAQ API.

## Using the Data Queue

If the data queue is specified properly, each time a spooled file is converted to PDF and stored in the integrated file system or document library system, an entry is sent to the queue. To use the data queue, follow these steps:

1. Use **Create data queue** (CRTDTAQ) to create the data queue with these values:
  - **Maximum message length** (MAXLEN) is 714 or higher.
  - **Sequence** (SEQ) is \*FIFO or \*LIFO.
2. Specify the name of the data queue on the PSF configuration object. To do this, use the PDFDTAQ parameter on the Create PSF configuration object (CRTPSFCFG) or Change PSF configuration object (CHGPSFCFG) command. PDFGEN must have a value other than \*NONE.



If a data queue's format is not valid and PSF tries to add entries to it, PSF sends a message to the message queue associated with the print writer and continues processing.

It is the user's responsibility to manage the data queues. This includes creating, clearing, and deleting the data queue.

## Format

Table 5 shows the format of an entry sent to the data queue when a spooled file is converted to PDF.

*Table 5. Data Queue Entry Format*

Decimal Offset	Hex Offset	Type	Description
0	0	CHAR(10)	Function
10	A	CHAR(2)	Record type
12	C	CHAR(2)	Return value
14	E	CHAR(26)	Qualified job name <b>CHAR(10)</b> Job name <b>CHAR(10)</b> User name <b>CHAR(6)</b> Job number
40	28	CHAR(10)	Spooled file name
50	32	CHAR(2)	Reserved
52	34	BINARY(4)	Spooled file number
56	38	CHAR(250)	Mail tag
306	132	CHAR(340)	Path and name of PDF file
646	286	CHAR(2)	Reserved
648	288	BINARY(4)	Path name CCSID
652	28C	CHAR(10)	Mail sender
662	296	CHAR(10)	User data
672	2A0	CHAR(80)	Reserved

A description of each field follows:

### Function

Identifies the record type within the function that created the data queue entry. The value for converting a spooled file to PDF is \*PDFWTR.

### Record type

Identifies the function that created the data queue entry. Valid values are:

**01** Spooled file conversion record

### Return value

Identifies whether the conversion from IPDS to PDF was successful. Values are:

**01** File was successfully converted and stored in specified folder.

**02** File was successfully converted and stored on specified output queue.

**03** File was successfully converted and sent as e-mail.

- 04 File was successfully converted but the PDF mapping program specified not to mail PDF file.
- 05 File was successfully converted but was not e-mailed because the mail tag or address was not valid.
- 06 Conversion error.

**Qualified job name**

Identifies the qualified job name of the job that created the spooled file that was converted to PDF. The first 10 characters are the job name, the next 10 characters are the user name, and the last six characters are the job number.

**Spooled file name**

Identifies the name of the spooled file that was converted to PDF.

**Spooled file number**

Identifies the unique number of the spooled file that was converted.

**Mail tag**

Identifies the mail address or keyword specified on the printer file or spooled file **User defined data** (USRDFNDDTA) parameter or the Group name. A Group name is specified by the DDS STRPAGGRP keyword, the CRTAFPDTA Group Name, or the AFP Toolbox Begin Group command. If you specify a keyword, such as a customer number, it must be mapped to a valid e-mail address by a PDF mapping program).

**Path and name of PDF file**

Identifies the path and name of the PDF file.

**Path name CCSID**

The CCSID of the path and PDF file name.

**Mail sender**

The name of the mail sender from **User defined data** or the PSF configuration object.

**User data**

The value in the printer file **User defined data** parameter. It contains blanks if **User defined data** contains blanks.

---

## Chapter 7. Transforming PCL, PDF, and PostScript Data to AFP

This chapter describes how to use Transform Manager to convert PCL, PDF, and PostScript to AFP. It contains this information:

- “Overview” describes how these transforms work.
- “Syntax” on page 116 gives the command syntax.
- “Planning for the PCL, PDF, or PostScript to AFP Transforms” on page 116 describes what you need to do before using the transforms.
- “Transforming Data” on page 122 explains how to use the transforms.
- “Font Downloader” on page 124 describes how to use the Font Downloader with the PostScript to AFP transform.
- “Troubleshooting” on page 126 lists some common problems and explains how to fix them.

---

### Overview

These transforms can convert the following data streams to image-based Advanced Function Presentation (AFP) format for printing on IBM AFP printers:

- Printer Control Language (PCL) 6
- Portable Document Format (PDF) 1.3
- PostScript Language Level 3

When a spooled file is released and has one of the data formats for which a transform is provided, an OS/400 AFP printer writer can automatically call image print transform. Next, image print transform calls the appropriate transform program. The PDF, PCL, or PostScript data is then converted to an AFP image to preserve fidelity and printed on the IPDS printer. If you want a printer writer to automatically use a transform, you must configure the printer writer appropriately. See “Setting Up the Printer Device Description” on page 122 for information about configuring the printer writer.

Each transform can output data suitable for printing on two types of printers: edge-to-edge and no-print border. Edge-to-edge printers have no set margin. No-print border printers have a set .167 inch margin.

Figure 16 on page 116 shows how data is transformed from PCL, PDF, or PostScript to AFP, after the transforms have been enabled on your OS/400:

1. A PCL, PDF, or PostScript job is sent to image print transform.
2. Image print transform calls the appropriate exit program.
3. The exit program requests a transform from Transform Manager and tells Transform Manager the image configuration object type. This tells Transform Manager what type of transform is needed.
4. When there is no transform process available, Transform Manager either starts a new transform process or waits for one to finish. For information about how Transform Manager determines when to start a new transform job, see “Configuring Transform Manager” on page 117.
5. Transform Manager starts an appropriate transform if necessary.
6. The exit program sends the PCL, PDF, or PostScript data to the transform.

7. The transform sends the transformed job to the exit program. If Transform Manager started this transform job to fulfil a request, the transform job ends.
8. The exit program sends the AFP data to image print transform.
9. Image print transform sends the AFP data to PSF.
10. PSF prints the image-based AFP data on an IPDS printer.

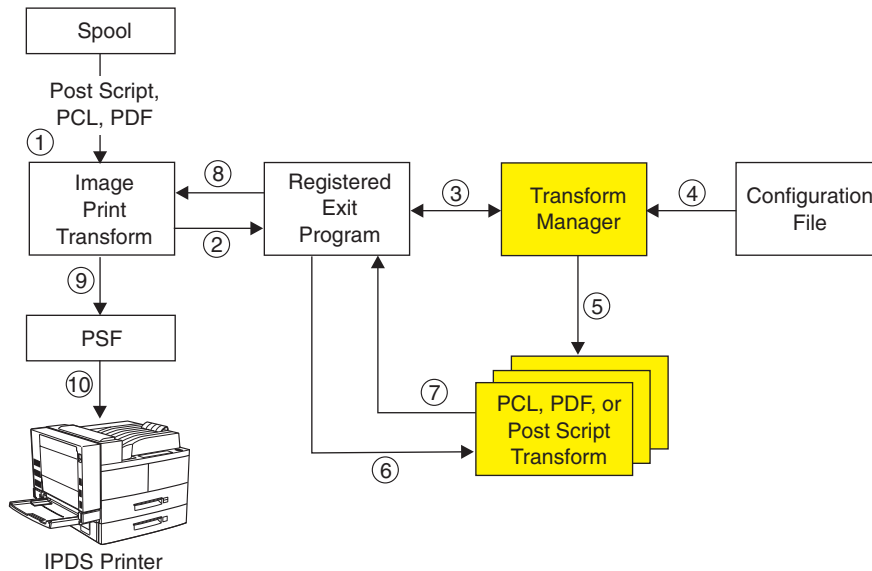


Figure 16. Transforming PCL, PDF, and PostScript data to AFP Data.

## Syntax

Figure 17 shows the command syntax used to start Transform Manager.

### Start Transform Manager

```
▶▶—STRTFMMGR—◀◀
```

Figure 17. Starting Transform Manager

Figure 18 shows the command syntax used to end Transform Manager.

### End Transform Manager

```
▶▶—ENDTFMMGR—◀◀
```

Figure 18. Ending Transform Manager

## Planning for the PCL, PDF, or PostScript to AFP Transforms

Before transforming PCL, PDF, or PostScript data, you need to do the following:

- Ensure that you have the necessary authority. See “Authorities” on page 117.
- Ensure that your system is able to run the transform. See “System Requirements” on page 117.
- Create the **work** directory. See “Configuring Transform Manager” on page 117.

- Configure Transform Manager. See “Configuring Transform Manager”.
- Customize the PostScript to AFP transforms (optional). See “Customizing the PostScript to AFP and PDF to AFP Transforms” on page 120.
- Start Transform Manager. See “Starting and Ending Transform Manager” on page 121.
- Set up the printer device description. See “Setting Up the Printer Device Description” on page 122.
- Increase the maximum memory setting - optional. See “Increasing the Maximum Memory Setting” on page 121.

## Authorities

To issue the STRTFMMGR command, you must have \*JOBCTL special authority.

## System Requirements

In order to use the PCL, PDF, or PostScript to AFP transform, you need a V5R1 or higher OS/400, with the PASE feature installed. If you issue the STRTFMMGR command on a system that does not have PASE installed, a message is issued and Transform Manager does not start.

## Creating the Work Directory

You must create the **work** directory with the proper authorities in order for Transform Manager to work properly. Create this path on your system with the specified owner and permissions:

/QOpenSys/QIBM/UserData/InfoprintServer/Transforms/work

Table 6. Directories to Create for Transform Manager

Directory	Owner	Permissions
InfoprintServer	QIPSJOB	755
Transforms	QIPSJOB	755
work	QIPSJOB	700

To change the directory information, follow these steps:

1. Call `qp2term` from the command line.
2. Specify `cd` to change to the directory that contains the directory you want to edit. For example, `cd /QOpenSys/QIBM/UserData/` allows you to edit the InfoprintServer directory.
3. Specify `chown new_owner directory` to change the directory owner. For example, `chown QIPSJOB InfoprintServer` makes QIPSJOB the owner of InfoprintServer.
4. Specify `chmod new_permissions directory` to change the directory permissions. For example, `chmod 755 InfoprintServer` makes 755 the permissions for InfoprintServer.

**Note:** There cannot be anything in the **work** directory.

## Configuring Transform Manager

When you configure Transform Manager, you can specify these:

- The number of each type of transform to initialize when Transform Manager is started.

- The maximum number of each type of transform that can be active.

For example, you can specify that Transform Manager starts five PCL edge-to-edge transforms upon startup, and that no more than ten PCL edge-to-edge transforms can run simultaneously. The appropriate values for minimum and maximum active depend on your print environment. To make an informed decision, you must understand how Transform Manager starts and ends transform jobs:

1. At startup, Transform Manager starts the specified minimum number of each type of transform.
2. When a writer gets a job that needs to be transformed, it tries to get control of an appropriate transform. There are several possible outcomes to this:
  - An appropriate transform is available. In this situation, the writer gives the job to that transform. The selected transform does not take any new jobs until it is done transforming the current job.
  - No appropriate transform is available and the maximum number of that type of transform has not been started. In this situation, Transform Manager starts the appropriate transform. The job waits while the transform starts, and after the job is transformed, the transform ends.
  - No appropriate transform is available and the maximum number of that type of transform has been started. In this situation, the writer waits until a transform becomes available or until the maximum is no longer reached.

You should avoid starting new transform jobs frequently, having lots of extra transforms active, and having writers frequently waiting for transforms. Specify **minimum active** and **maximum active** so they reflect your estimated transform needs.

To configure Transform Manager, follow these steps:

1. Copy the configuration file **qxtrtfmmgr.cfg** from **/QOpenSys/QIBM/ProdData/InfoprintServer/Transforms/** into **/QOpenSys/QIBM/UserData/InfoprintServer/Transforms/**.  
IBM recommends that you copy the file by entering the following on the command line: CPY  
OBJ ('/QOpenSys/QIBM/ProdData/InfoprintServer/Transforms/qxtrtfmmgr.cfg')  
TOOBJ ('/QOpenSys/QIBM/UserData/InfoprintServer/Transforms/qxtrtfmmgr.cfg').  
You can also use iSeries Navigator to copy the file.
2. Specify appropriate values for **minimum active** and **maximum active**. Use the EDTF command from the OS/400 command line to edit the configuration file. Blank lines and lines that start with a pound sign (#) are ignored.

**Notes:**

- a. The values specified for **PS Transforms** apply to both PostScript and PDF transforms.
- b. The configuration file is encoded in **EBCDIC** and must remain so to obtain the expected results.
- c. The CCSID of the configuration file must be 037. The CCSID might be unintentionally changed by using the copy commands from Windows or from the qp2term shell. You can use iSeries Navigator to verify the CCSID.
- d. Do not change anything in the configuration file except the values for minimum active and maximum active.

Figure 19 on page 119 shows the default configuration file.

```
#
# Infoprint Server Default Transform Manager Configuration file
#

PS Transforms:
image config type = EdgeToEdge
minimum active = 1
maximum active = 1

PCL Transforms:
image config type = EdgeToEdge
minimum active = 1
maximum active = 1

PS Transforms:
image config type = NoPrintBorder
minimum active = 1
maximum active = 1

PCL Transforms:
image config type = NoPrintBorder
minimum active = 1
maximum active = 1
```

*Figure 19. Default Configuration File*

Keywords:

### **PS Transforms**

This tag identifies the start of a PostScript transform configuration block. Two PostScript configuration blocks are required; one for edge-to-edge jobs and one for no-print border jobs.

### **PCL Transforms**

This tag identifies the start of a PCL transform configuration block. Two PCL configuration blocks are required; one for edge-to-edge jobs and one for no-print border jobs.

### **image config type**

This field identifies the type of image configuration object that is specified on the device description of the printer writer. Jobs from this writer either go to an edge-to-edge transform or one with a no-print border. The printer that the image configuration object is for determines which. See “Setting Up the Printer Device Description” on page 122 for more information about the image configuration object.

### **minimum active**

This field specifies the number of transform processes that Transform Manager starts, and the minimum number of transform processes that Transform Manager keeps active. Specify a value in the range 0-32767. The default is 1.

### **maximum active**

This field specifies the maximum number of transform processes that Transform Manager makes active. When this number is reached, Transform Manager does not start a new transform process to perform a transform. Therefore, the transform request waits until a transform process is available. This value must be greater than the value specified for **minimum active**. Specify a value in the range 1-32767. The default is 1.

### **Example**

The following example specifies the following:

- No transforms are initialized to process data for PCL, PDF, PostScript edge-to-edge jobs, but one transform can start.
- At least five, but no more than 10 transforms are started to process PostScript and PDF data for jobs with a no-print border.
- At least three, but no more than 10 transforms are started to process PCL data for jobs with a no-print border.

PS Transforms:

```
image config type = EdgeToEdge
minimum active = 0
maximum active = 1
```

PCL Transforms:

```
image config type = EdgeToEdge
minimum active = 0
maximum active = 1
```

PS Transforms:

```
image config type = NoPrintBorder
minimum active = 5
maximum active = 10
```

PCL Transforms:

```
image config type = NoPrintBorder
minimum active = 3
maximum active = 10
```

## Customizing the PostScript to AFP and PDF to AFP Transforms

You might want to customize the PDF and PostScript to AFP transform in one of these ways:

- “Adding Fonts” lets you print documents with fidelity when the needed font is on your system.
- Increasing the Maximum Memory Setting. Do this if you are experiencing memory errors when transforming PDF files to AFP. See page 121.

### Adding Fonts

You can add fonts to the PostScript to AFP and PDF to AFP transforms. This lets you print documents with fidelity when the needed font is on your system. To add fonts to the transform edit the font mapping file. The font mapping file is encoded in ASCII.

To edit the font mapping file, create the file **fonts.map** in this path **/QOpenSys/QIBM/UserData/InfoprintServer/Transforms/ps2afp/fonts**. Add any fonts to this file that are not mapped in **fonts.map** in the ProdData path: **/QOpenSys/QIBM/ProdData/InfoprintServer/Transforms/ps2afp/fonts**. The format of each entry in the file is:

```
font name path
```

*name* specifies a font name and *path* specifies the path where it is stored. For example, these are some of the entries in **font.map** in the ProdData path:

```
font BookMaster /QOpenSys/QIBM/ProdData/InfoprintServer/Transforms/ps2afp/fonts/BookMaster
```

```
font BookMaster-Bold /QOpenSys/QIBM/ProdData/InfoprintServer/Transforms/ps2afp/fonts/BookMaster-Bold
```

```
font BookMaster-BoldItalic /QOpenSys/QIBM/ProdData/InfoprintServer/Transforms/ps2afp/fonts/BookMaster-BoldItalic
```



When a new transform is started, Transform Manager checks for **fonts.map** in the UserData path. If it is not found, only the **fonts.map** in the ProdData path is used. If it is found, the file in the UserData directory is checked for a font first. If the font is not mapped there, the **fonts.map** file in the ProdData path is searched.

**Notes:**

1. Ensure that the path in **fonts.map** points to the correct font file.
2. The data in **fonts.map** in the UserData path takes precedence over the data in **fonts.map** in the ProdData path. If the same font is mapped in both places, the path specified in **fonts.map** in the UserData path is used.
3. Only a system administrator should change **fonts.map**. If there are incorrect entries in **fonts.map**, Transform Manager will not start.

### Increasing the Maximum Memory Setting

Some incoming PDF data streams require more memory than others to transform successfully. The amount of memory required is affected by the size of the incoming PDF file and, more importantly, the way the PDF datastream was constructed. By default, 24M of memory is provided for the PDF and PostScript to AFP transform. If you suspect that your jobs are failing with memory failures, increase the maximum amount of memory available for the transform. This increase of memory should not affect other portions of the system.

IBM recommends that you increase the limit to the maximum value allowed (99M) to verify that you are experiencing memory failures. If the error is corrected by increasing the memory limit, IBM recommends that you leave the memory limit at the maximum value since there is minimum effect of keeping the limit at the maximum level.

To increase the maximum limit for the PDF and PostScript to AFP transform, change the default configuration file. This change will affect all subsequent instances of the PostScript and PDF transform.

To modify the configuration file for the PDF and PostScript to AFP transform, follow these steps:

1. From the OS/400 command line, enter:  
`edtf '/QOpenSys/QIBM/ProdData/InfoprintServer/Transforms/ps2afp/ps2afpd.cfg'`
2. Change the line: `ps_max_memory = 24M` to `ps_max_memory = 99M`.
3. Save and exit the edit session on the file.
4. If Transform Manager is running, from the OS/400 command line enter:  
`ENDTFMMGR`
5. Before transforming any PCL, PDF, or PostScript files to AFP, enter:  
`STRTFMMGR`

## Starting and Ending Transform Manager

Enter STRTFMMGR on the command prompt to start Transform Manager. This starts the minimum number of each type of transform specified. See “Configuring Transform Manager” on page 117 for information about specifying the minimum number of transforms. You can only run one Transform Manager job at a time. If Transform Manager is already running and you issue this command, the new job is not started. A message is then sent to the job log and the screen.

Enter ENDTFMMGR on the command prompt to end Transform Manager. This command ends all transform jobs started by Transform Manager and ends any active jobs from printer writers.

## Setting Up the Printer Device Description

These transforms are called when a PCL, PDF, or PostScript job is sent to a printer writer that has the transforms enabled. To enable the transforms, the printer device descriptions must be set up correctly. The device must be configured to use AFP data, and must have the image print transform enabled. The image print transform function is enabled automatically when the device has an image configuration object defined. For every writer that will call these transforms, specify these values in the device description:

- AFP(\*YES)
- IMGCFG(\*IMGCnn), nn is in the range 01-11.

Transform Manager uses the information from the image configuration object associated with the printer writer to call the appropriate transform. The image configuration objects specify to which type of IPDS printer the transformed data is sent. See Table 7 to determine the appropriate image configuration object.

Table 7. Image Configuration Objects

Image Configuration Object	Dots Per Inch	Edge-to-Edge or No-Print Border	Other
*IMGC01	240	Edge-to-Edge	MMR
*IMGC02	300	Edge-to-Edge	MMR
*IMGC03	600	Edge-to-Edge	MMR
*IMGC04	1200	Edge-to-Edge	MMR
*IMGC05	240	No-Print Border	MMR
*IMGC06	300	No-Print Border	MMR
*IMGC07	600	No-Print Border	MMR
*IMGC08	1200	No-Print Border	MMR
*IMGC09	240	Edge-to-Edge	IM/1 image
*IMGC10	240	No-Print Border	IM/1 image
*IMGC11	240	Edge-to-Edge	CCITT G4 compression

For information about image print transform or image configuration objects, refer to *Printer Device Programming* or the Redbook *IBM AS/400 Printing V*. You can obtain Redbooks from the Online Publications Website <http://publib.boulder.ibm.com/>.

---

## Transforming Data

To use these transforms directly from your OS/400, release a PCL, PDF, or PostScript job to a printer writer that has the transforms enabled. The printer writer must be properly configured. See “Setting Up the Printer Device Description” on page 122 for information about configuring the printer writer.

You can use these transforms from your personal computer in three ways:

- Send a job on your workstation to a shared PCL, PDF, or PostScript printer. For information about setting up a printer share on the OS/400, refer to *iSeries Guide to Advanced Function Presentation*.
- Use LPR to print the data on your OS/400-attached IPDS printer. When you use LPR, you must submit an appropriate options file with the data, or your output will be incorrect. For information about using LPR and an option file, refer to *TCP/IP Configuration and Reference*.

- Call the transforms from an application program by using the Convert Image (QIMGCVTI, QimgCvtImg) API. For information about using this API, refer to the iSeries Information Center.

## Deactivating and Reactivating the Transforms

If Infoprint Server is installed on your system and you specified an appropriate image configuration object and AFP(\*YES) in the device description, these transforms are invoked any time you send PCL, PDF, or PostScript data to the device. If you do not want these transforms invoked and do not want to uninstall Infoprint Server, you must delete the exit programs that Infoprint Server registers.

By default, Infoprint Server registers three exit programs to the the QIBM\_QIMG\_TRANSFORMS exit point for the PCL, PDF, and PostScript to AFP transforms:

Table 8. Exit Programs Created by Infoprint Server

Exit Program Number	Exit Program	Library
5380	QXTRCLIENT	QIPS
5381	QXTRCLIENT	QIPS
5382	QXTRCLIENT	QIPS

## Steps to Deactivate the Transforms

Follow these steps to deactivate the Infoprint Server PCL, PDF, and PostScript to AFP transforms.

1. On the command line, enter WRKREGINF and press F4. Enter option 8 (work with exit programs) by the QIBM\_QIMG\_TRANSFORMS exit point and press enter:

```

Work with Registration Information
Type options, press Enter.
  5=Display exit point  8=Work with exit programs
Exit
Point
Opt  Point      Format  Registered  Text
QIBM_AIA_RETR_INF  RTVI0100 *YES   BRM Services/400 object retri
QIBM_AIA_TAPE_INF  MEDI0100 *YES   BRM Services/400 media inform
QIBM_AIA_TAPE_MOVE MEDM0100 *YES   BRM Services/400 media moveme
QIBM_QCA_CHG_COMMAND CHGC0100 *YES   Change command exit programs
QIBM_QCA_RTV_COMMAND RTVC0100 *YES   Retrieve command exit program
QIBM_QCQ_AGENT     ENDE0100 *YES
QIBM_QCQ_AGENT     STRE0100 *YES
QIBM_QGW_NJEOUTBOUND NJEO0100 *YES   Network Job Entry outbound ex
QIBM_QHQ_DTAQ      DTAQ0100 *YES   Original Data Queue Server
8  QIBM_QIMG_TRANSFORMS XFRM0100 *YES
QIBM_QJO_DLT_JRNRCV DRCV0100 *YES   Delete Journal Receiver
More...

Command ==>
F3=Exit  F4=Prompt  F9=Retrieve  F12=Cancel

```

2. The Work with Exit Programs display opens. Use option 4 (Remove) to remove the three exit programs shown:

```

                                Work with Exit Programs
Exit point:  QIBM_QIMG_TRANSFORMS      Format:  XFRM0100
Type options, press Enter.
  1=Add   4=Remove   5=Display   10=Replace
      Exit
      Program      Exit
Opt      Number      Program      Library
  4          5380      QXTRCLIENT  QIPS
  4          5381      QXTRCLIENT  QIPS
  4          5382      QXTRCLIENT  QIPS

                                Bottom
Command ==>
F3=Exit   F4=Prompt   F5=Refresh   F9=Retrieve   F12=Cancel

```

## Reactivating the Transforms

To reactivate the PCL, PDF, and PostScript to AFP transforms, you must re-register the exit programs by entering these commands.

**Important:** Each part of the program data (PGMDTA) must be eight bytes long. For example, for PCL to AFP exit program, the program data is 'PCL+5 spaces+AFPDS+3 spaces'.

```

ADDEXITPGM EXITPNT(QIBM_QIMG_TRANSFORMS) FORMAT(XFRM0100) PGMNBR(5380) PGM (QIPS/QXTRCLIENT)
THDSAFE(*NO) MLTTHDACN(*MSG) TEXT('IBM InfoprintServer PCL, PDF and PS TO AFP')
CRTEXITPNT(*YES) PGMDTA(37 16 'PCL      AFPDS  ')

ADDEXITPGM EXITPNT(QIBM_QIMG_TRANSFORMS) FORMAT(XFRM0100) PGMNBR(5381) PGM (QIPS/QXTRCLIENT)
THDSAFE(*NO) MLTTHDACN(*MSG) TEXT('IBM InfoprintServer PCL, PDF and PS TO AFP')
CRTEXITPNT(*YES) PGMDTA(37 16 'PS      AFPDS  ')

ADDEXITPGM EXITPNT(QIBM_QIMG_TRANSFORMS) FORMAT(XFRM0100) PGMNBR(5382) PGM (QIPS/QXTRCLIENT)
THDSAFE(*NO) MLTTHDACN(*MSG) TEXT('IBM InfoprintServer PCL, PDF and PS TO AFP')
CRTEXITPNT(*YES) PGMDTA(37 16 'PDF      AFPDS  ')

```

## Font Downloader

Infoprint Server Font Downloader (hereafter referred to as Font Downloader) lets you use certain double-byte fonts with the PostScript to AFP transform. These fonts are downloaded from a Mac to your PC with one of these client products: IBM Infoprint Manager for Windows (Program Number 5639-N49), Infoprint DBCS Font Downloader feature (LCD4-5884-00) or IBM Infoprint Manager for AIX (Program Number 5765-E42), Infoprint DBCS Font Downloader feature (LCD4-5884-00). Font Downloader then uploads the fonts to your OS/400. Hereafter, the client products are referred to as Infoprint DBCS Font Downloader client.

When a Font Downloader job is started, it listens at a specified port for a connection from an Infoprint DBCS Font Downloader client application. When a new font is received, the Font Downloader job puts the font in the correct directory. It then modifies the font mapping table and sets up symbolic links to the downloaded font directory. This is illustrated in Figure 20 on page 125.

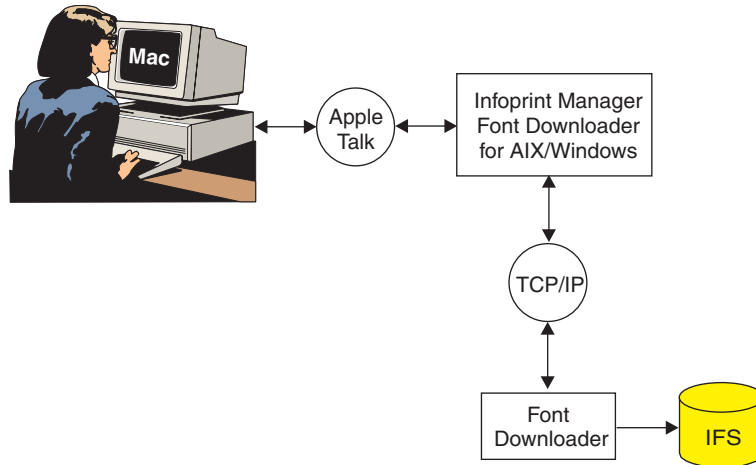


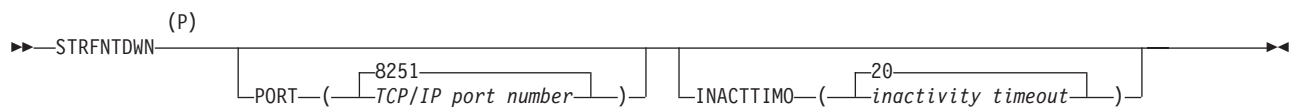
Figure 20. Font Downloader Process

**Security Note:** As shown in Figure 20, Font Downloader opens an iSeries TCP/IP port. No user authentication is required to connect to this port. For this reason, two additional security features have been added. First, the iSeries font installation job that is started by STRFNTDWN has a time-out feature. By default, the job ends and the TCP/IP port is closed after 20 minutes. Second, the font installation job runs under the QIPJOB profile. QIPJOB owns all fonts installed on the iSeries server.

## Syntax

Figure 21 shows the command syntax used to start Font Downloader.

### Start Font Downloader



#### Notes:

P All parameters preceding this point can be specified in positional form.

Figure 21. Starting Font Downloader

Figure 22 shows the command syntax used to end Font Downloader.

### End Font Downloader



Figure 22. Ending Font Downloader

### Parameters

You can use the following parameters to customize the Start Font Downloader command.

#### PORT

Specifies on which TCP/IP port the Font Downloader job should listen for

Infoprint DBCS Font Downloader client connections. You will get an error if you try to start Font Downloader on a port that is already being used. Therefore, you should avoid using commonly used port numbers.

**8251** Font Downloader listens on port 8251.

*TCP/IP-port-number*

Font Downloader listens on the specified port number.

## INACTTIMO

Specifies how long (in minutes) the Font Downloader job is inactive before ending.

**20** Font Downloader ends if there is no activity between Font Downloader and the Infoprint DBCS Font Downloader client for 20 minutes.

*inactivity-timeout*

Font Downloader ends if there is no activity between Font Downloader and the Infoprint DBCS Font Downloader client for the specified number of minutes. Specify a value in the range 1-9999 minutes.

## Examples

The following command starts a Font Downloader job if one is not already running. The job listens on port 6001 for requests from the Infoprint DBCS Font Downloader client. If no requests are received from the client in ten minutes, the Font Downloader job is ended.

```
STRFNTDWN PORT(6001) INACTTIMO(10)
```

The following command ends a Font Downloader job if one is running.

```
ENDFNTDWN
```

## Troubleshooting

This section describes some common errors, what could cause them, and possible ways fix them.

Table 9. Troubleshooting the Transforms Managed by Transform Manager

Problem	Possible Cause	Possible Solutions
The spooled file stays in PND status.	There are no available transforms to process the request. Either many requests are queued or a very large job is being transformed. One or more transforms might have ended abnormally or might be hung.	In the configuration file, increase the minimum active, the maximum active value, or both, then issue ENDTFMMGR followed by STRTFMMGR.
The spooled file cannot be printed and error CPCnnnn is issued.	The exit point registration information is incorrect or corrupted.	Replace the existing information with the correct values. See Appendix D, "Exit Points" on page 163 for instructions about viewing and changing exit point QIBM_QIMG_TRANSFORMS information.

Table 9. Troubleshooting the Transforms Managed by Transform Manager (continued)

Problem	Possible Cause	Possible Solutions
Transform Manager does not start.	<ul style="list-style-type: none"> <li>Your system does not meet the minimum requirements.</li> <li>You do not have authority to issue the command.</li> <li>The configuration file is incorrect.</li> <li>The <b>fonts.map</b> file contains an entry that does not exist.</li> </ul>	<ul style="list-style-type: none"> <li>Ensure that your system meets the minimum requirements to run Transform Manager. See “System Requirements” on page 117.</li> <li>Ensure that you have *JOBCTL special authority.</li> <li>Correct the configuration file <b>qxtrtfmmgr.cfg</b>. See “Configuring Transform Manager” on page 117 for the proper format.</li> <li>Check all of the <b>fonts.map</b> entries. See “Customizing the PostScript to AFP and PDF to AFP Transforms” on page 120.</li> </ul>
Document prints with wrong orientation	The wrong paper size is specified in the printer file.	Correct the printer file. Do not use paper size to specify orientation; this information is contained in the data. For example, to print a document on 11 X 17 paper in landscape orientation, specify 11 X 17 paper, not 17 X 11.
A PDF file does not transform correctly. The print request produces error pages that report problems with the pdfprint function and returns a PostScript error page.	The PDF file requires more memory to transform.	Increase the maximum memory setting for the PDF to AFP transform. See “Increasing the Maximum Memory Setting” on page 121 for instructions.





## Chapter 8. Sending E-mail

The PDF subsystem lets you transform a spooled file to PDF and e-mail it. You can use a specified SMTP mail server or the OS/400 Send Distribution (SNDDST) command to send the e-mail. The input spooled file can be any type of file that PSF can print. The e-mail function is flexible, so you can use it to accomplish both simple and complex tasks. For example, you can use it to send one file to one recipient, or divide one file into many files and send each file to a different recipient.

**Note:** The PDF subsystem interacts with PSF to transform data and send e-mail. However, you do not need a PSF for OS/400 license to use the PDF subsystem.

With a PDF mapping program you can resolve mail tags, specify the subject text, and add a customized message to the beginning of each e-mail. If you use an SMTP server to send the e-mail, you can also use a PDF mapping program to specify carbon copy (cc), blind carbon copy (bcc), and return-to addresses, a file to use as the e-mail body, and attachments.

This chapter explains how to use the e-mail function, how to recover from errors, and how to use the PDF mapping program. The table below lists the tasks associated with sending e-mail and specifies whether they are required to send e-mail with Infoprint Server.

*Table 10. E-mail Tasks*

Task	Page	Required
Enabling Your OS/400 to Send E-Mail	130	Yes
Setting up the OS/400 as a POP E-mail Client	132	No
Creating a PSF Configuration Object	133	Yes
Configuring the Device	96	Yes
Varying the Device ON	96	Yes
Starting the Writer	97	Yes
Using Mail Tags	133	No
Using the PDF Mapping Program	143	No*
Modifying the Input Data	99	No
Using AFP Manager to Send E-mail	135	No
Checking for Non-delivery Messages	136	No
Setting QUTCOFFSET	137	No

\* Required if you are using mail tags that are not valid e-mail addresses.

**Note:** When using an SMTP server to send e-mail, PSF must establish a connection with the server each time it wants to send an e-mail. For example, if you want to send 10 e-mails, PSF must establish a connection with the SMTP server 10 times.

---

## Steps to Send a PDF File as E-mail

To send a PDF file as e-mail, follow these steps or use AFP Manager to access the PDF subsystem. For information about using AFP Manager to access the PDF subsystem, see “Accessing the PDF Subsystem with iSeries Access” on page 98.

### Before you begin:

You must have all of these to successfully send an e-mail:

- An OS/400 system enabled to send e-mail. See “Enabling Your OS/400 to Send E-Mail”.
- An OS/400 system set up as a POP server (optional, but required to receive e-mail). See “Setting up the OS/400 as a POP E-mail Client” on page 132.
- An appropriate PSF configuration object varied ON. See “Creating a PSF Configuration Object” on page 133 and “Varying the Device ON” on page 96.
- A properly configured device started. See “Configuring the Device” on page 96 and “Starting the Writer” on page 97.

### Steps to send a PDF file as e-mail:

1. Modify the input file to add mail tags (optional). See “Using Mail Tags” on page 133.
2. Create a PDF mapping program (optional). See Appendix B, “PDF Mapping Program” on page 143.
3. Print the IPDS file on a properly configured device.

---

## Enabling Your OS/400 to Send E-Mail

Infoprint Server can convert a spooled file to PDF and send it as e-mail. The input spooled file can be any type of data that PSF can print. To send a file as e-mail, your OS/400 must be properly configured.

**Note:** These instructions are for setting up your OS/400 to send outgoing e-mail. If you want to e-mail PDF documents to local OS/400 user accounts, you must also set up your system to receive e-mail. See “Setting up the OS/400 as a POP E-mail Client” on page 132 for information about setting up your system to receive e-mail.

These instructions are for users in an environment that meet all of these requirements:

- TCP/IP is configured.
- E-mail has not been set up before.
- You want to use the command line to configure your system for sending e-mail.
- You want to send e-mail with Infoprint Server.

If you use iSeries Navigator or Operations Navigator, you can follow the steps in the iSeries Information Center to set up your system to send e-mail. To find these instructions, go to Networking → TCP/IP → E-mail → Configure e-mail.

## Steps to Enable Your OS/400 to Send E-mail

These are the steps required to set up your OS/400 to send e-mail with Infoprint Server 5.1 and higher. Enter the commands with the given parameters and values as shown in capital letters below. If you prefer, enter the command (such as CHGDSTA), then press F4 to be prompted through the command. Pressing F11

toggles between the parameter keywords (such as SMTPRTE) and a list (sometimes partial) of acceptable values for that parameter.

1. Specify the user ID and address that the mail services will use to route mail that has an internet address as recipients.

This is not the ID of an actual person. For this example, we use INTERNET as the user ID and GATEWAY for the address. We will create a directory entry with this user ID and address in step 2. On the command line, enter:

```
CHGDSTA SMTPRTE(INTERNET GATEWAY)
```

**Restriction:** You need security administrator (\*SECADM) authority to use this command.

2. Create a directory entry that the mail services will use to route mail that has an internet address as recipients.

User ID and Address must be the same values you specified for User ID and Address in step 1. On the command line, enter:

```
ADDIRE USRID(INTERNET GATEWAY) USRD('Generic Internet user') SYSNAME(INTERNET)
      PREFADR(NETUSRID *IBM ATCONXT)
```

3. Specify that outgoing e-mail messages are not split apart and specify that the POP mail servers are started when TCP/IP is started. On the command line, enter:

```
CHGPOPA MSGSPLIT(*NOMAX) AUTOSTART(*YES)
```

4. Configure the SMTP server to specify that it is started when TCP/IP is started and to specify the mailrouter to use to get through the Firewall, if you have one. If you exchange e-mail with Domino servers, specify \*NO for Percent routing character (PCTRTGCHR) along with the parameters listed below. On the command line, enter:

```
CHGSMTPA AUTOSTART(*YES) MAILROUTER('mailrouter') FIREWALL(*YES)
```

A value for MAILROUTER is required only if you want to control what happens when a name and address cannot be resolved. The mail router name would typically be set up by the system administrator. Specify MAILROUTER(\*NONE) if you are unsure.

5. Create user profiles for any users who want to send e-mail. We will create user JOEJ. Enter:

```
CRTUSRPRF USRPRF(JOEJ) PASSWORD(not2bad) TEXT('Joe Johnson')
```

6. Add users to the system distribution directory. The value for USER must be the same as the value for USRPRF specified in step 5. To use this ID to send e-mail, specify this user ID in the PDFSENDER parameter of the PSF configuration object used with the PDF subsystem. Alternatively, you can specify it in USRDFNDDTA in the spooled file's attributes or the printer file. ISERIES is the system name of the iSeries that Joe works on. Enter:

```
ADDIRE USRID(JOEJ ISERIES) USRD('Joe Johnson') USER(JOEJ)
```

7. **Note:** Only complete this step if you want to use an SMTP server to send e-mail with Infoprint Server. You cannot specify an SMTP server with Infoprint Server 5.1. This step requires \*SECADM special authority.

Add an SMTP address for the users in the system distribution directory.

- a. To add an SMTP address for the user ID JOEJ, enter WRKDIRE and press Enter.
- b. Choose option 2 (Change) by the user ID you want to change and press Enter.
- c. Press F19 (Add name for SMTP). If there is no SMTP address associated with this ID, a message appears on the bottom of the screen. Press Enter.

- d. Specify either the SMTP user ID and domain or the SMTP route. We specify the user ID and domain of Joe's Internet e-mail address: JOEJ@SUPERSUNSEEDS.COM.

Add Name for SMTP

System: ISERIES

Type choices, press Enter.

User ID . . . . . : JOEJ

Address . . . . . : ISERIES

SMTP user ID . . . . . JOEJ

SMTP domain . . . . . SUPERSUNSEEDS.COM

SMTP route . . . . .

F3=Exit F4=Prompt F12=Cancel

8. Stop the TCP/IP server. Enter `ENDTCPSVR SERVER(*SMTP)`

9. Start the TCP/IP server. Enter `STRTCPSPVR SERVER(*SMTP)`

You can now test the new e-mail setup by using the `SNDDST` command. We suggest you do this before testing Infoprint Server for the first time.

10. Test the e-mail setup by entering this command. You can use the `TOUSRID` or `TOINTNET` parameters or both:

```
SNDDST TYPE(*LMSG) TOUSRID((user address)) TOINTNET((me@mysystem.com)) DSTD('E-mail')
      LONGMSG('This is a test to determine whether my new e-mail setup works.')
      SUBJECT('E-mail test')
```

11. Go to your Internet mail system or OS/400 to see if you have received the message. For instructions about using `SNADS` to receive e-mail on your OS/400, refer to the iSeries Information Center.

For more information about these commands, refer to the OS/400 online help, the iSeries Information Center, or the OS/400 Service Web Page at:  
<http://www.ibm.com/servers/eserver/iseries/service>

---

## Setting up the OS/400 as a POP E-mail Client

You must set up your OS/400 as a POP e-mail client if you want to do any of these:

- Receive e-mail
- Receive error messages when your mail cannot be delivered by another server
- Use your OS/400 as an SMTP server to send e-mail

For instructions, refer to "Setting up POP e-mail clients" in the iSeries Information Center.

---

## Creating a PSF Configuration Object

You need a PSF configuration object to tell PSF to transform the spooled file to PDF and send it as e-mail. To create a PSF configuration object for sending e-mail, enter this command:

```
CRTPSFCFG PSFCFG(name_of_configuration_object) PDFGEN(*MAIL)
```

You can use the PDFMAILSVR, PDFSENDER, and PDFMAPPGM parameters to further customize the e-mail characteristics. For information about the parameters you can specify with CRTPSFCFG, see “Setting up Your PSF Configuration Object” on page 95 and the CRTPSFCFG Command Description topic in the iSeries Information Center.

You can use AFP Manager, a component of Client Access Express 5.1 and iSeries Access 5.2, to create and change PSF configuration objects. For information about using AFP Manager to work with PSF configuration objects, refer to *AS/400 Guide to Advanced Function Presentation and Print Services Facility*.

---

## Using Mail Tags

A mail tag specifies where the e-mail is sent. It can be a valid e-mail address (in the form *name@domain*) or it can be a keyword, like a customer number. If the mail tag is a keyword, use a PDF mapping program to interpret it. You can put mail tags in the printer file, in the data, or in both. If you specify a PDF mapping program, all of the tags will be interpreted by the PDF mapping program. If you do not specify a PDF mapping program, all mail tags must be valid e-mail addresses.

To use mail tags you must complete these tasks:

- Put the mail tags in the printer file or input file.
- Create a PDF mapping program if the mail tags are not valid e-mail addresses.
- Specify the PDF mapping program, if one is used, on the PSF configuration object.

## Inserting a Mail Tag in the Printer File

To add a mail tag in the printer file *file-name*, enter this command:

```
CHGPRTF FILE(library file-name)  
  USRDFNDA('MAILTAG(tag-name)MAILSENDER(mail-sender-name)')
```

The tag name can be up to 80 characters long. If the mail tag is not a valid e-mail address, you must specify a PDF mapping program that converts the mail tag to an e-mail address. Specify the PDF mapping program on the PSF configuration object PDFMAPPGM parameter. For information about using a PDF mapping program, see Appendix B, “PDF Mapping Program” on page 143 and Appendix C, “Templates” on page 155.

MAILSENDER is optional. If you use it, MAILSENDER overrides the PDFSENDER parameter on the PSF configuration object. The mail sender name can be up to 10 characters long. MAILSENDER must be a valid user profile and must have a directory entry in the system distribution directory. For using SMTP mail servers, you must have an SMTP address in the system distribution directory. Use Display Directory Entries (DSPDIRE) to view the entry in the System Distribution Directory and Work with Directory Entries (WRKDIRE) to change an entry. For more detailed information about working with the System Distribution Directory, see “Enabling Your

OS/400 to Send E-Mail” on page 130. For information about using a printer file, refer to *AS/400 Guide to Advanced Function Presentation and Print Services Facility*.

## Inserting a Mail Tag in the Spooled File

You can add a mail tag or specify the mail sender after the file is spooled. If you do this, it overrides any value previously specified for MAILTAG or MAILSENDER in the printer file. To add a mail tag to the file after it has been spooled, enter this command:

```
CHGSPLFA FILE(file-name) JOB(fully-qualified-job-name) SPLNBR(spooled-file-number)  
USRDFNDA('MAILTAG(tag-name)MAILSENDER(mail-sender-name)')
```

Alternatively, you can follow these steps:

1. Enter WRKSPLF.
2. Specify option 2 "Change Spooled File Attributes" by the spooled file you want to change.
3. Press F9 "All parameters".
4. Change the User defined data (USRDFNDA) parameter:  
USRDFNDA('MAILTAG(*tag-name*)MAILSENDER(*mail-sender-name*)')

The tag name can be up to 80 characters long. If the mail tag is not a valid e-mail address, you must specify a PDF mapping program that converts the mail tag to an e-mail address. Specify the PDF mapping program on the PSF configuration object PDFMAPPGM parameter. For information about using a PDF mapping program, see Appendix B, "PDF Mapping Program" on page 143 and Appendix C, "Templates" on page 155.

MAILSENDER is optional. If you use it, MAILSENDER overrides the PDFSENDER parameter on the PSF configuration object. The mail sender name can be up to 10 characters long. MAILSENDER must be a valid user profile and must have a directory entry in the system distribution directory. For using SMTP mail servers, you must have an SMTP address in the system distribution directory. Use Display Directory Entries (DSPDIKE) to view the entry in the System Distribution Directory and Work with Directory Entries (WRKDIKE) to change an entry. For more detailed information about working with the System Distribution Directory, see "Enabling Your OS/400 to Send E-Mail" on page 130.

## Inserting a Mail Tag in the Input File

For DDS print applications, you can put a mail tag on the DDS STRPAGGRP keyword. See "Using DDS to Modify the Input File" on page 99 for information about using DDS to insert a mail tag in the input file. For more information about using DDS, refer to *DDS Reference*.

You can use the CRTAFPDTA indexing function to generate multiple PDF files and e-mail them. For instructions, see "E-mailing the CRTAFPDTA Output" on page 32.

The AFP Toolbox Begin Group command generates appropriate group tags for use with the PDF subsystem. The Group name can specify a mail tag. See "Using AFP Toolbox to Modify the Input File" on page 100 for information about using AFP Toolbox to insert a mail tag in the input file. For more information about AFP Toolbox, refer to *AFP Toolbox for Multiple Operating Systems User's Guide*.

---

## Using AFP Manager to Send E-mail

AFP Manager is a plug-in to iSeries Navigator, a component of iSeries Access 5.2 or Operations Navigator, a component of Client Access 5.1. Unless otherwise specified, iSeries Navigator refers to both iSeries Navigator and Operations Navigator. With iSeries Access 5.2, you can use AFP Manager to select spooled files to convert to PDF and send as e-mail. If you want to send the e-mail to many people, you should use mail tags and a PDF mapping program because of performance considerations.

### Before You Begin

- Make sure that you have a PSF configuration object that tells PSF to transform the spooled file into a PDF file.

There are several ways to create a new PSF configuration object or modify an existing one:

- To use AFP Manager to create a PSF configuration object, follow these steps:
    1. Double-click the iSeries Access icon.
    2. Double-click the iSeries Navigator icon.
    3. Open a connection to a 5.1 or higher OS/400.
    4. Expand **AFP Manager**.
    5. Right-click **PSF Configurations** and select **New...**
    6. On the PDF Transform page, select **Generate PDF** and **Send as electronic mail**. For help on any field, select the field and press F1.
    7. Click **OK**.
  - Run the command Create PSF Configuration (CRTPSFCFG) or Change PSF Configuration (CHGPSFCFG) from the OS/400 command line. Specify PDFGEN (\*MAIL). You can specify other appropriate values as described in “Setting up Your PSF Configuration Object” on page 95.
  - Run the command Create PSF Configuration (CRTPSFCFG) or Change PSF Configuration (CHGPSFCFG) from iSeries Navigator. Specify PDFGEN (\*MAIL). You can specify other appropriate values as described in “Setting up Your PSF Configuration Object” on page 95. Refer to the iSeries Information Center or *AS/400 Guide to Advanced Function Presentation and Print Services Facility* for steps to run an OS/400 command from iSeries Navigator.
- Make sure that the writer that processes the spooled files is configured properly. To configure a writer to transform a spooled file to PDF, run the Create Device Description, Printer (CRTDEVPRT) or Change Device Description, Printer (CHGDEVPRT) command. Specify the values listed in “Configuring the Device” on page 96. You can use iSeries Navigator or the OS/400 command line to run commands. Refer to the iSeries Information Center or *AS/400 Guide to Advanced Function Presentation and Print Services Facility* for steps to run an OS/400 command from iSeries Navigator.

### Steps to Create a PDF File and Send it as E-mail

1. In iSeries Navigator, expand **My Connections** (or your active environment).
2. Expand a 5.2 OS/400 system.
3. Expand **Basic Operations** and select **Printer Output**.
4. Right-click the output files you want to convert and select **Convert to PDF...** . The **Convert Printer Output to PDF** dialog opens with the specified printer output files in the **Printer output** list.



5. Select **Send as electronic mail**.
6. Specify whether to use the PDF mapping program specified on the PSF configuration object. If you use the PDF mapping program, PSF sends the e-mail to all recipients specified in the PDF mapping program as well as the recipients specified in the **To:** field.
7. Specify the device that PSF should use to convert the printer output file. You can specify the library-qualified output queue, the printer whose queue is going to do the conversion, or both.
8. Use the **To:** field to specify where the e-mail is sent. Separate addresses with a space or comma, or put them on separate lines.
9. Click **OK**.

A new spooled file is created for each spooled file you selected. For help with any field, select the field and press **F1**.

---

## Checking for Non-delivery Messages

When an e-mail is sent and either the destination user name or the domain is incorrect, a message is returned to the sender. The sender name can be specified with the PDFSENDER parameter in the PSF configuration object that the PDF subsystem uses, or it can be set in the printer file or the spooled file by specifying a value for MAILSENDER in USRDFNDDTA. The spooled file setting overrides the PSF configuration object. For information about other errors when sending e-mail, see "Error Recovery" on page 110.

Some thought should be given as to how you set up and manage the sender user ID for the PDF subsystem. If you set up your PSF configuration object with the default value of PDFSENDER(\*SPLFOWN), you might have to enroll many different users so they can receive non-delivery messages. For business critical applications, consider setting up a special user ID for the e-mail sender and have one person responsible for checking for messages and taking appropriate action, such as contacting the intended recipient and re-sending the document.

Use the PSF configuration object PDFMAILSVR parameter to specify how the e-mail is sent. You can specify the SNDDST command or an SMTP server. If you are using SMTP to send the e-mail and a client is configured to receive e-mail for the e-mail sender, no extra setup is required.

The SNDDST command does not have a specific vehicle to check for these incoming messages, but they are accessible and should be monitored. To receive non-delivery messages when you specify that SNDDST is used to send the e-mail, set up a Post Office Protocol Version 3 (POP3) client associated with the sender's user ID. To do this, map the sender's entry in the system distribution directory to an SMTP address, such as *name@domain*.

## Setting up a POP3 Client for a SNDDST Sender

1. Enroll the sender in the System Directory. See step 6 on page 131.
2. Associate the sender's user ID with an SMTP user ID. See step 7 on page 131.
3. Configure a client to receive email for that user. There are many software products that can be used to act as a client to the iSeries POP3 server. These include clients such as Eudora and Netscape. Configuration steps depend on the product's individual interface. However, the information you provide is basically the same.

**Example:**



This example uses Netscape Mail as the client.

- a. Open a Netscape Navigator window.
- b. Expand the **Edit** menu and select **Preferences**.
- c. In the left pane expand **Mail & Newsgroups** and select Mail Servers.
- d. Click **Add**.
- e. On the **General** page, enter these values:

In the **Server Name** field, enter the SMTP domain that you specified in step 2 on page 136.

In the **Server Type** field, select POP3 Server.

In the **User Name** field, enter the user ID you created for the e-mail sender.

When the e-mail sender retrieves mail, the user is prompted for the iSeries password for that user ID.

---

## Setting QUTCFFSET

The time stamp placed into the e-mail is in Greenwich Mean Time (GMT) unless the system value QUTCFFSET (Coordinated Universal Time Offset) is set according to your geographic location. If QUTCFFSET is not set correctly, the time stamp might be incorrect. These are some of the appropriate values:

*Table 11. Appropriate Time Offsets*

City	Offset
London	+00:00
Montreal	-05:00
Riyadh	+03:00
San Francisco	-08:00
Tokyo	+09:00



---

## Chapter 9. Related Products

This chapter describes these products that are related to Infoprint Server:

- “iSeries Access”
- “iSeries Access for Web”
- “Infoprint Designer for iSeries” on page 140

---

### iSeries Access

This section describes what iSeries Access is and how to obtain it. For instructions to complete tasks related to Infoprint Server, refer to *iSeries Guide to Advanced Function Presentation*, the iSeries Information Center, or the iSeries Navigator online help.

#### Overview

iSeries Access is IBM's Windows client for connectivity to the iSeries. It is a separately orderable program (Program Number 5769-XE1). For ordering information see

<http://www.ibm.com/servers/eserver/iseries/clientaccess/caorder.htm>.

Unless otherwise noted, iSeries Access refers to both Client Access Express and iSeries Access.

You can use iSeries Access to accomplish a wide variety of tasks including these:

- Print data generated from a user's PC application on an IPDS printer - uses iSeries Navigator.
- Create PC resources as resources on the OS/400 - uses iSeries Navigator.
- Create or change a PSF configuration object - uses iSeries Navigator.
- Access the PDF subsystem - uses iSeries Navigator. This lets you convert a spooled file to PDF then send it as e-mail, store it in the integrated file system, or re-spool it. See “Accessing the PDF Subsystem with iSeries Access” on page 98 for instructions.
- Run any OS/400 command - uses iSeries Navigator.
- View AFP files - uses AFP Viewer.

---

### iSeries Access for Web

iSeries Access for Web (Program Number 5722XH2 for 5.2) gives you access to your iSeries from a Web browser such as Netscape, Opera, or Internet Explorer. This product runs on Windows, Linux, and AIX systems. However, all of the necessary code is on the iSeries server, so you do not have to install anything on the client. iSeries Access for Web lets you complete these tasks and more:

- View and control printers, internet printers, printer shares, spooled files (view as GIF, TIFF, PCL, or AFP with AFP viewer), and output queues.
- Display or send OS/400 messages, view message queues.
- Access the OS/400 file system and file shares.
- Run CL commands with prompt support.

- View spooled files in PDF. This uses Infoprint Server if it is installed. If not, the file is transformed to TIFF, which is then converted to PNG and embedded in a PDF document. For instructions, refer to *iSeries Guide to Advanced Function Presentation*.
- Copy, delete, and rename files.

For information about iSeries Access for Web, refer to this Web page:  
<http://www.ibm.com/servers/eserver/iseries/clientaccess/web/>.

---

## Infoprint Designer for iSeries

Infoprint Designer for iSeries (Program Number 5733-ID1) is available for OS/400 4.5 or higher. Infoprint Designer provides a new way for you to format your data and create AFP overlays and page segments. It has a true WYSIWYG (what you see is what you get) interface. This lets you see exactly how the formatted application will look while you work on it. The package includes overlay creation and data layout functions which are closely integrated, plus a separate Image Editor program for producing and editing page segments.

For more information about Infoprint Designer, refer to *IBM Infoprint Designer for iSeries: Getting Started* and the Redbook *IBM eserver iSeries Printing VI: Delivering the Output of e-business*.

---

## Appendix A. Related Tasks

---

### Turning a PC AFP Resource into an OS/400 Resource

You can turn an AFP resource (page segment or overlay) on your PC into an AFP resource on your OS/400 without iSeries Access by following these steps.

1. Make sure that the PC AFP resource is in `\\system\QDLS\`, where *system* is the name of your OS/400. If the AFP resource is not in that directory, map a network drive to your OS/400 system then use Windows Explorer to move the file.
2. On your OS/400 system, create a physical file. You will copy the resource into this physical file.
  - Specify `CRTPF FILE(library/file_name) RCDLEN(8201)` on the command line. *file\_name* and *library* are the name of the new physical file and the library you want it created in, respectively.
3. Copy the Windows AFP resource into this new physical file:
  - Specify `CPYFRMPCD FROMFLR(folder) TOFILE(library/file_name) FROMDOC(resource_name) TRNTBL(*NONE)` on the command line. *folder* specifies the name of a folder or folder path containing the PC resource that is copied. *file\_name* and *library* are the same values you specified in step 2 and *resource\_name* is the name of the PC AFP resource.
4. Create an OS/400 object from the physical file:
  - To create a page segment, specify `C RTPAGSEG PAGSEG(pagseg_library/pagseg_name) FILE(library/file_name) MBR(file_name)` on the command line. *pagseg\_name* and *pagseg\_library* specify the name and library of the page segment that is created.
  - To create an overlay, specify `CRTOVL OVL(ovl_library/ovl_name) FILE(library/file_name) MBR(file_name)` on the command line. *ovl\_name* and *ovl\_library* specify the name and library of the overlay that is created.
  - In the above, *file\_name* and *library* are the same values you specified in step 1.

For instructions about using iSeries Access to do this, refer to *AS/400 Guide to Advanced Function Presentation and Print Services Facility*.



---

## Appendix B. PDF Mapping Program

The PDF mapping program is an exit program that lets you map a keyword, for example, a customer number, to one or more e-mail addresses and an optional mail message. You can also use the PDF mapping program to specify the subject text and text to include at the beginning of each e-mail. If you use an SMTP server to send the e-mail, you can also use the PDF mapping program to specify carbon copy (cc), blind carbon copy (bcc), and return-to addresses, a file to use as the e-mail body, and attachments.

If specified, the PDF mapping program is called for each successfully generated PDF file when the PDFGEN parameter has a value of \*MAIL. Once PSF receives notification from the virtual printer that all pages have been stacked, it calls the PDF mapping program for each PDF file. The PDF mapping program then resolves the mail tags before mailing the file. It does not check the format of the mail tag, but tries to map each tag.

When you print to a device specified as AFP(\*YES) and a PSF configuration object that specifies PDFGEN(\*MAIL), you can specify a PDF mapping program. Use the PDFMAPPGM parameter on the CRTPSFCFG or CHGPSFCFG command to specify the program. For more information about e-mailing PDF files, see “Setting up Your PSF Configuration Object” on page 95 and “Using Mail Tags” on page 133. A sample PDF mapping program is available in Appendix C, “Templates” on page 155.

Infoprint Server provides a data queue where notifications of PDF transform completion can be stored. Your PDF mapping program can use the Receive Data Queue API (QRCVDTAQ) to access this information. For more information about the data queue, see “PDF Conversion Completion Data Queue” on page 112. For more information about using APIs, refer to *System API Programming* or go to the iSeries Information Center.

The following sections describe various fields and formats that can be used in the PDF mapping program. However, not all fields can be used when PSF uses the SNADS protocol to send your e-mail. PSF uses the SNADS protocol when PDFMAILSVR(\*SNDDST) is specified. The fields that can be used with SNADS are marked in Table 12. All fields can be used when an SMTP mail server is specified in PDFMAILSVR.

Table 12. Fields that Can be Used with SNADS

Field Name	Format Name	SNADS-Yes
CCSID of message text data and subject	Base	
Disposition of PDF file	Base	X
More processing	Base	X
Email addresses	Base	
Length of BCC email addresses	Extension area	
Length of CC email addresses	Extension area	
Length of email address data	Base	X
Length of Extension area format	Extension area	X

Table 12. Fields that Can be Used with SNADS (continued)

Field Name	Format Name	SNADS-Yes
Length of individual stream file format	Individual stream file information	
Length of message text	Base	X
Length of path name	Individual stream file information	
Length of path name for directory for files	Extension area	
Length of ReplyTo email address	Extension area	
Length of subject	Extension area	X
Message text data	Base	X
Number of stream files specified	Stream file	
Offset to BCC email addresses	Extension area	
Offset to CC email addresses	Extension area	
Offset to extension area	Base	X
Offset to list of path names for body of e-mail	Extension area	
Offset to list of path names of attachments	Extension area	
Offset to path name	Individual stream file information	
Offset to path name for directory for files	Extension area	
Offset to ReplyTo email address	Extension area	
Offset to subject	Extension area	X
Stream file information	Stream file	
Total length for stream file information	Stream file	
Total length of individual stream file format	Individual stream file information	
Use specified directory	Individual stream file information	

## Parameters

This section describes the PDF mapping program input and output parameters. Table 13 lists the required parameters for the PDF mapping program.

Table 13. Required Parameter Group

Parameter	Input/Output	Type
Email exit input information	Input	Char(*)
Length of input information	Input	Binary(4)
Email exit output information	Output	Char(*)
Length of output information buffer	Input	Binary(4)
Length of output information available	Output	Binary(4)



### E-mail exit input information

The PDF mapping program information that is input to the PDF mapping program from the PSF print writer. For the format description of the information, see Table 14.

### E-mail exit output information

The PDF mapping program information which is output from the PDF mapping program to the PSF print writer. For the format description of the information, see Table 15 on page 147.

### Length of input information

The length of the input information.

### Length of output information available

You set this value to signify you have more data than will fit in the **length of output information buffer**. This will be the actual size you need to fit all your data into. If this length is less than or equal to the length of the mail tag output information, all available information has been returned. If this length is greater than the length of the mail tag output information, data beyond the length is truncated. Only data up to the length provided is returned in the e-mail exit output information buffer. The **length of output information buffer** specifies the maximum amount of data that can be returned by the user.

### Length of output information buffer

The size, in bytes, of the mail tag output information buffer. This field is the length of the buffer that a you have to contain your data. If it is not large enough to contain all the data, use the **length of output information available** field to tell PSF how large of a buffer you need. Then PSF will call the PDF mapping program again with the enlarged buffer size. See “PDF Mapping Program in C” on page 155 or “PDF Mapping Program in RPG” on page 156 for an example.

## E-mail Input information

This section describes the input to the PDF mapping program. Table 14 shows the structure for the e-mail input parameter.

Table 14. E-mail Input Information Structure

Decimal Offset	Hex Offset	Type	Description
0	0	CHAR(26)	Qualified job name
26	1A	CHAR(10)	Spooled file name
36	24	BINARY(4)	Spooled file number
40	28	CHAR(250)	Mail tag
290	122	CHAR(340)	Path and name of PDF file
630	276	CHAR(1)	Mail server type
631	26C	CHAR(1)	Reserved
632	26E	BINARY(4)	Path and name CCSID
636	272	CHAR(10)	Mail sender
646	27C	CHAR(10)	User data (USRDTA)
656	290	CHAR(8)	Job system name
664	298	CHAR(8)	Creation time stamp

Table 14. E-mail Input Information Structure (continued)

Decimal Offset	Hex Offset	Type	Description
672	2A0	CHAR(10)	Output queue on which the spooled file is located
682	2AA	CHAR(10)	Output queue library

A more detailed description of each entry in the table follows in alphabetical order according to Description:

**Creation time stamp**

Identifies the date and time when the spooled file was created. See the QWCCVTDT API and format \*DTS for more information.

**Job system name**

Identifies the name of the system on which the spooled file was created.

**Mail sender**

Identifies the sender of the e-mail. This value is specified on the spooled file or printer file USRDFNDDTA parameter or on the PSF configuration object PDFSENDER parameter. If the value is specified in all three places, the value that is specified on spooled file attributes is used. If no mail sender is identified on USRDFNDDTA, the value from the PDFSENDER parameter of the PSF configuration object is used.

**Mail server type**

Identifies the mail server to use. Values are:

- '1'      Use the Send Distribution (SNDDST) command.
- '2'      Use the SMTP protocol to send e-mail.

**Mail tag**

Identifies the mail address or mail tag specified on the spooled file or printer file USRDFNDDTA parameter or the Group tag in the data. This should be blanks ('40'X) if no value was provided by either parameter. If both provide tag or address information, the Group tag takes precedence.

The PDF mapping program can return mail addresses based on job, file, and USRDTA information.

**Output queue library**

Identifies the library where the above output queue is located.

**Output queue on which the spooled file is located**

Identifies the output queue to which the writer has been started.

**Path and name CCSID**

The CCSID of the path and name of the PDF file.

**Path and name of the PDF file**

Identifies the path and name of the PDF file.

**Qualified job name**

Identifies the qualified job name of the job that created the spooled file that was converted to PDF. The 26 characters are specified as follows:

**CHAR(10)**

Job name

**CHAR(10)**

User name

**CHAR(6)**

Job number

**Spooled file name**

Identifies the name of the spooled file that was converted to PDF.

**Spooled file number**

Identifies the unique number of the spooled file that was converted to PDF.

**User data**

The value in the printer file USRDTA parameter. This contains blanks if USRDTA contains blanks.

## E-mail Output

This section describes the output for the PDF mapping program. Table 15 shows the structure for the e-mail output parameter. Note that each byte of any reserved field in the e-mail output structure must be initialized to X'00'.

*Table 15. PDF Mapping Program Output Information Structure*

Decimal Offset	Hex Offset	Type	Description
0	0	CHAR(1)	Disposition of the PDF file
1	1	CHAR(1)	More processing
2	2	CHAR(2)	Reserved (set each byte to X'00')
4	4	BINARY(4)	Length of message text
8	8	BINARY(4)	Length of mail address
12	C	CHAR(255)	Message text
267	10B	CHAR(1)	Reserved (set to X'00')
268	10C	BINARY(4)	Offset to extension area
272	110	BINARY(4)	CCSID of message text and subject
276	114	CHAR(11)	Reserved (set each byte to X'00')
287	11F	CHAR(*)	E-mail address

A more detailed description of each entry in the table follows in alphabetical order according to Description:

**CCSID of message text and subject**

The CCSID associated with the message text and subject, if specified. This CCSID is used to convert the message text and subject to Unicode before they are placed in the e-mail. The e-mail receiver's e-mail program might have to be configured for the appropriate character set in order to view the e-mail correctly.

This field is only used when an SMTP mail server is specified in your PSF configuration object. If you are not using an SMTP mail server, set this field to 0. Valid values are:

**0** Use the job's CCSID.

*ccsid-value*

Specifies the CCSID. Values are in the range 1-65533.

### **Disposition of PDF file**

- '0' Do not mail PDF file
- '1' E-mail PDF file to specified addresses.

### **E-mail addresses**

The e-mail addresses (in the form *name@domain*) to which the PDF file is sent. They must be delimited by single quotation marks, for example, ('name1@domain1' 'name2@domain2')

### **Length of e-mail address**

The length of the e-mail address data to be used to send the PDF file. The maximum length allowed is 16MB because this is the largest value that can be used to allocate a user space. If the PDF file should not be e-mailed, set the length of the e-mail address data to zero. The initial amount of space that is allowed for e-mail addresses is 49 bytes. If you need more space than that, use the PDF mapping program to extend the size. "Exit Program in C" on page 159 gives an example of this situation.

### **Length of message text**

The length of the message text to be used in the body of the e-mail. Values can be in the range 0-255. If no message text is to be used when sending the PDF file, set the length of the message text data to zero.

### **Message text**

The message text data to be used in the body of the e-mail when sending the PDF file. This value should be blanks if the text length is zero.

You can specify both Message text and files for the body of the e-mail. If you do not specify Message text or a file to be used for the body of the e-mail, the default text is used. The default text is obtained from message PQT4133 in message file QPQMSGF.

PSF skips a line after the message text. Two end-of-line characters are written after your data. Any files specified for inclusion in the body of the e-mail are positioned at this point.

If you specify an SMTP mail server, the CCSID of the message text and subject is used to convert the data to Unicode before adding it to the e-mail. The e-mail receiver's e-mail program might have to be configured for the appropriate character set in order to view the e-mail correctly.

### **More processing**

Specifies whether the PDF mapping program should be called again to do more processing for the same spooled file.

#### **'0' (X'F0')**

Do not call the PDF mapping program again. This field occupies the offset of a field that was reserved in Infoprint Server 5.1. Therefore, X'00' is treated the same as 0 (X'F0') for compatibility.

#### **'1' (X'F1')**

Call the PDF mapping program again. The input information presented to the PDF mapping program is not modified. If an error occurs at any point while processing an e-mail, the spooled file is held and the PDF mapping program is not called again, regardless of this field.

### **Offset to extension area**

Offset from the beginning of this structure. The value must be a 4-byte multiple.

A value of 0 indicates that the extension area is not being used. See “Extension Area Format” for details about the extension area.

## Extension Area Format

This section describes the format of the extension area. The extension area is only specified when needed. If you do not need it, specify 0 for Offset to extension area in the PDF Mapping Program Output Information, described in Table 15 on page 147.

This format must begin on a 4-byte multiple. All offsets are computed from the beginning of the PDF Mapping Program Output Information structure.

Table 16. Extension Area Format

Decimal Offset	Hex Offset	Type	Description
0	0	BINARY(4)	Length of Extension area format
4	4	BINARY(4)	Offset to subject
8	8	BINARY(4)	Length of subject
12	C	BINARY(4)	Offset to ReplyTo e-mail address
16	10	BINARY(4)	Length of ReplyTo e-mail address
20	14	BINARY(4)	Offset to CC e-mail address
24	18	BINARY(4)	Length of CC e-mail address
28	1C	BINARY(4)	Offset to BCC e-mail address
32	20	BINARY(4)	Length of BCC e-mail address
36	24	BINARY(4)	Offset to list of path names for body of e-mail
40	28	BINARY(4)	Offset to path name for directory for files
44	2C	BINARY(4)	Length of path name for directory for files
48	30	BINARY(4)	Offset to list of path names of attachments

A more detailed description of each entry in the table follows in alphabetical order according to Description:

### Length of BCC e-mail address

Specifies the length of the BCC e-mail address to send the e-mail to. This field is only used when an SMTP mail server is specified in your PSF configuration object. If an SMTP mail server is not specified, set this field to 0.

### Length of CC e-mail address

Specifies the length of the CC e-mail address to send the e-mail to. This field is only used when an SMTP mail server is specified in your PSF configuration object. If an SMTP mail server is not specified, set this field to 0.

### Length of Extension area format

Specifies the length of the Extension area format. This value must be set to 52.

### Length of path name for directory for files

Specifies the length of the path name containing the directory in which files can be stored. This field is only used when an SMTP mail server is specified in your PSF configuration object. If an SMTP mail server is not specified, set this field to 0.

### Length of ReplyTo e-mail address

Specifies the length of the ReplyTo e-mail address. This field is only used when an SMTP mail server is specified in your PSF configuration object. If an SMTP mail server is not specified, set this field to 0.

### Length of subject

Specifies the length of the character string to be used as the subject text. The maximum length is 256. Data exceeding the maximum length is truncated. If there is no subject, set this field to 0. If you do not specify subject text, default text is used.

### Offset to BCC e-mail address

Specifies the offset to the BCC e-mail addresses. This area contains the BCC e-mail addresses to which the PDF file is sent. The addresses must be delimited by single quotation marks. For example, ('name1@domain1' 'name2@domain2') Specify 0 if you do not want the e-mail sent to any BCC addresses.

### Offset to CC e-mail address

Specifies the offset to the CC e-mail addresses. This area contains the CC e-mail addresses to which the PDF file is sent. The addresses must be delimited by single quotation marks. For example, ('name1@domain1' 'name2@domain2') Specify 0 if you do not want the e-mail sent to any CC addresses.

### Offset to list of path names for body of e-mail

Specifies the offset to the list of path names for the files to include in the body of the e-mail. You can specify one or more files to include, but the files must be in the integrated file system. If you include the PDF file you are sending in this list, it will be attached again. Specify 0 if you do not want any files included in the body of the e-mail. This field is only used when an SMTP mail server is specified in your PSF configuration object. If an SMTP mail server is not specified, set this field to 0.

The CCSID of the file is used if a code page conversion is required. See “Stream File Format” on page 152 for information about the format to use when specifying this data.

#### Notes:

1. The order in which you list these files is the order in which they are included in the e-mail. These files are included after the message text, if specified.
2. You must specify a thread-safe file system. PSF can only access thread-safe file systems. These file systems are not thread-safe:
  - QNetware
  - QFileSvr.400
  - Network File System (NFS)
  - QDLS
3. All files must be encoded in a CCSID that matches the CCSID of the target system. For example, if you are using a that contains the Euro symbol in CCSID 923, ensure that the receiving system uses CCSID923.
4. If a file is not text (file type .txt) or HTML (file type .htm or .html), it is included as an attachment to the e-mail. For example, if you specify a sound file to be part of the e-mail body, it is attached to the e-mail instead.
5. If you indicate that this file is to be used with the **path name for directory for files**, the file name is appended to the **path name for directory for files**. Do not use a “/” as the first character when specifying the file name. However, the file can be nested within another directory, such as

mySubDirectory/myFile.txt. For example, if you specified /MyCompany/AllFiles as the **path name for directory for files**, PSF would search for myfile.txt in /MyCompany/AllFiles/mySubDirectory/myfile.txt.

6. If you indicate that this file is not to be used with the **path name for directory for files**, then the file name must be fully-qualified, beginning with a “/”.

#### Offset to list of path names of attachments

Specifies the offset to the list of path names for the files to be attachments to the e-mail. You can specify one or more files to attach, but the files must be in the integrated file system. Do not include the original PDF file in this list unless you want the file to be attached twice. Specify 0 if you do not want any files (other than the original PDF file) attached. This field is only used when an SMTP mail server is specified in your PSF configuration object. If an SMTP mail server is not specified, set this field to 0.

If you specified a directory to be used for files (see **Offset to path name for directory for files**), then the path name specified for this field is appended to the value specified for the directory when you specify a '1' for the file in **Use specified directory** parameter in the “Individual Stream File Information Format” on page 153. If you did not specify a directory to be used for files, the path name specified for each attachment must be fully resolved.

See “Stream File Format” on page 152 for information about the format to use when specifying this data.

#### Notes:

1. You must specify a thread-safe file system. PSF can only access thread-safe file systems. These file systems are not thread-safe:
  - QNetware
  - QFileSvr.400
  - Network File System (NFS)
  - QDLS
2. If the attachments are to be viewed by the receiver, make sure that they are in a CCSID that matches the CCSID of the target system. For example, if you are using a system that contains the Euro symbol in CCSID 923, ensure that the receiving system uses CCSID923. Files encoded in EBCDIC are not viewable.
3. If you indicate that this file is to be used with the **path name for directory for files**, the file name is appended to the **path name for directory for files**. Do not use a “/” as the first character when specifying the file name. However, the file can be nested within another directory, such as mySubDirectory/myFile.txt. For example, if you specified /MyCompany/AllFiles as the **path name for directory for files**, PSF would search for myfile.txt in /MyCompany/AllFiles/mySubDirectory/myfile.txt.
4. If you indicate that this file is not to be used with the **path name for directory for files**, then the file name must be fully-qualified, beginning with a “/”.

#### Offset to path name for directory for files

Specifies the offset to a path name containing the directory in which files can be stored. This path name must be fully resolved, but if the path name does not end with a “/”, PSF adds that character. If you use this field, path names of files specified with **Offset to list of path names for body of e-mail** and **Offset to list of path names of attachments** are appended to the path name specified at the offset for this field. The path names of files must not begin with a “/”.



However, the file can be nested within another directory, such as mySubDirectory/myFile.txt. For example, if you specified /MyCompany/AllFiles as the **path name for directory for files**, PSF would search for myfile.txt in /MyCompany/AllFiles/mySubDirectory/myfile.txt.

This field is only used when an SMTP mail server is specified in your PSF configuration object. If an SMTP mail server is not specified, set this field to 0. A value of 0 indicates that no path name is specified. If you specify 0, you must specify the complete path name for all files used in the body of the e-mail and as attachments.

To use this directory, specify 1 in the field **Use specified directory** in the Individual stream file information format (see "Individual Stream File Information Format" on page 153). This format is used when specifying files to be used as attachments or as part of the e-mail body. If you specify 0 for **Use specified directory**, then that particular file must have a fully-specified path.

#### **Offset to ReplyTo e-mail address**

Specifies the offset to the ReplyTo e-mail address. Use this address to specify where replies to your e-mail are sent. This can be a different e-mail address than is specified for the FROM address on the PSF configuration object. The address must be in this form: ('email address').

A value of 0 indicates that no ReplyTo address is specified. This field is only used when an SMTP mail server is specified in your PSF configuration object. If an SMTP mail server is not specified, set this field to 0.

#### **Offset to subject**

Offset to the character string to be used as the subject text. A value of 0 indicates that a subject is not specified here. If you do not specify a subject in this structure, then the default subject is used. The default subject is obtained from message PQT4133 in message file QPQMSGF.

## **Stream File Format**

This section describes the format to use when specifying stream files in the Extension Area Format. Only use this format when you are specifying stream files to be included in the body or to be attached to the e-mail. The information that needs to be specified for stream files is listed in Table 17. You can only specify stream files when an SMTP mail server is specified in your PSF configuration object.

**Note:** The stream files must reside in the integrated file system. In order to use the stream files, the authority to the directories in the path where the stream file resides must be \*X. The authority to the stream files must be \*R.

*Table 17. Stream File Information*

<b>Decimal Offset</b>	<b>Hex Offset</b>	<b>Type</b>	<b>Description</b>
0	0	BINARY(4)	Total length of stream file information
4	4	BINARY(4)	Number of stream files specified
8	8	CHAR(*)	Stream file information

Detailed information about the fields follows, in alphabetical order according to Description:

#### **Number of stream files specified**

Indicates the number of stream files that are specified within this format.



**Stream file information**

Specifies the initial offset at which information for each stream file is positioned. See “Individual Stream File Information Format” for information about the format to use when specifying this information. Use one instance of the Individual stream file information format for each stream file. Each instance of this format must be contiguous to the previous one.

**Total length of stream file information**

Specifies the total length of all data used with this structure. This field is included in the length. The specified value must be a 4-byte multiple.

**Individual Stream File Information Format**

This section describes the format to use when specifying stream files. Table 18 lists the required format. Use one instance of the Individual stream file information format for each stream file. Each instance of this format must be contiguous to the previous one.

*Table 18. Stream File Information Format*

Decimal Offset	Hex Offset	Type	Description
0	0	BINARY(4)	Total length of individual stream file format
4	4	BINARY(4)	Length of individual stream file format
8	8	BINARY(4)	Offset to path name
12	C	BINARY(4)	Length of path name
16	10	CHAR(1)	Use specified directory
17	11	CHAR(3)	Reserved (set each byte to X'00')

A detailed description of each field follows, in alphabetical order according to Description:

**Length of individual stream file format**

Specifies the length of the structure used for the Individual stream file format. This value must be 20.

**Length of path name**

Specifies the length of the path name that specifies the stream file.

**Offset to path name**

Specifies the offset to the path name that specifies the stream file. The offset is from the beginning of the Individual stream file information format in which this value is specified.

**Total length of individual stream file format**

Specifies the total contiguous length used with this format, including this field. This value is used to determine the location of the next stream file information format and must be a 4-byte multiple.

**Use specified directory**

Specifies whether the directory specified for stream files (see **Offset to path name for directory for files** in “Extension Area Format” on page 149) is to be prefixed to the name of the stream file. For example, if you specify /SpecialApp/customerOne/ for a directory of files in the Extension area format, this value is prefixed to stream file names when 1 is specified in this field.

- |                   **'0'**     Do not prefix the directory to this path. The path name must begin  
|                   with a "/", otherwise, PSF will search for the file in the home  
|                   directory for user QSPLJOB.
- |                   **'1'**     Prefix the directory to this path. The path name must not begin with  
|                   a "/", since it is being appended to the specified directory.

## Appendix C. Templates

This appendix contains these templates:

- “PDF Mapping Program in C”
- “PDF Mapping Program in RPG” on page 156
- “Exit Program in C” on page 159

---

### PDF Mapping Program in C

This is a template for a PDF mapping program, it is an illustration intended to explain how to use a PDF mapping program. Before using this program, you must customize it for your environment. A PDF mapping program is an exit program that helps you customize the PDF subsystem. For information about using the PDF subsystem, see Chapter 6, “Using the PDF Subsystem” on page 93. For information about the exit program, see Appendix B, “PDF Mapping Program” on page 143.

```
| /*-----*/
| /*---                                          ---*/
| /*--- Program Name: ipstaggpm                ---*/
| /*---                                          ---*/
| /*--- Program Description: This is the template for a PDF mapping program.  ---*/
| /*---                                          ---*/
| /*---                                          ---*/
| /*--- Invocation: See calling convention below.  ---*/
| /*---                                          ---*/
| /*---                                          ---*/
| /*---                                          ---*/
| /*-----*/
| /*-----*/
| /*---                                          ---*/
| /*--- Calling convention:                    ---*/
| /*--- Pointer to E-mail exit input information. (char *, defined by Qpq_email_Exit_I_t ---*/
| /*---                                         in epqmapxt.h) ---*/
| /*--- Pointer to length of input information. (int *) ---*/
| /*--- Pointer to E-mail exit output information. (char *, defined by Qpq_Email_Exit_O_t ---*/
| /*---                                         in epqmapxt.h) ---*/
| /*---                                          ---*/
| /*--- Pointer to length of output information buffer (int *) ---*/
| /*--- Pointer to length of output information available (int *) ---*/
| /*---                                          ---*/
| /*-----*/
| /*-----*/
| /*--- Includes                               ---*/
| /*-----*/
| #include stdio.h
| #include string.h
| #include ctype.h
| #include stdlib.h
| #include epqmapxt.h
| /*-----*/
| /*--- Defines                               ---*/
| /*-----*/
| #define MAIL_ADDR_DATA_LENGTH 8096 /* Can be defined as needed*/
| /*-----*/
| /*--- Structures                            ---*/
| /*-----*/
| typedef struct output {
|     Qpq_Email_Exit_O_t exitOutputInfo;
|     char emailAddrData[MAIL_ADDR_DATA_LENGTH];
| } exitOutput ;
| int main (argc, argv)
|     int argc;
```

```

| char *argv[];
| {
|     exitOutput *output;
|     Qpq_Email_Exit_I_t *inputInfo;
|     /*-----*/
|     /*--- The following effectively makes the char[] fields preceding these two ---*/
|     /*--- fields in the structure able to be treated as null terminated strings. ---*/
|     /*-----*/
|     output = (exitOutput *)argv[3];
|     for (int i = 0; i < 2; i++) output->exitOutputInfo.Reserved1[i] = '\0';
|     output->exitOutputInfo.Reserved2[0] = '\0';
|     for (i = 0; i < 11; i++) output->exitOutputInfo.Reserved3[i] = '\0';
|     inputInfo = (Qpq_Email_Exit_I_t *) argv[1];
|     /*-----*/
|     /*--- Here is where the Mail Tag is read and resolved to e-mail addresses, ---*/
|     /*--- PDF Disposition, and message text data. Here is an example matching a tag ---*/
|     /*--- called TAG001 with a user at e-mail address 'ibmuser@us.ibm.com'. It sets ---*/
|     /*--- the disposition to 1 and sends a message. The message text data appears ---*/
|     /*--- as the body of the e-mail and the PDF file is sent as an attachment. If ---*/
|     /*--- no message text data specified, default text is used. ---*/
|     /*-----*/
|     if(strstr( inputInfo->Mail_Tag,"TAG001") != NULL)
|     {
|         strcpy(output->Email_addresses," 'ibmuser@us.ibm.com' ");
|         strcpy( output->Message_text_data,"This is a test message from IBMUSER");
|         output->PDF_File_Disposition[0] = '1';
|         len = strlen( output->Email_addresses);
|         output->Mail_address_data_length = 100;
|         output->Call_Exit_Program_Again[0] = '0';
|     }
| }
| /* end main */

```

---

## PDF Mapping Program in RPG

This is a template for a PDF mapping program, it is an illustration intended to explain how to use a PDF mapping program. Before using this program, you must customize it for your environment. A PDF mapping program is an exit program that helps you customize the PDF subsystem. For information about using the PDF subsystem, see Chapter 6, “Using the PDF Subsystem” on page 93. For information about the exit program, see Appendix B, “PDF Mapping Program” on page 143.

```

| D INPUTDS          DS
| D  JOBNAM          1      26
| D  SPLFID          27     36
| D  SPLNO           37     40B 0
| D  MAILTAG         41     290
| D  PDFFILE         291    630
| D  SVRTYPE         631    631
| D  RES1            632    632
| B D  PATHCCSID      633    636B 0
|   D  SENDER        637    646
|   D  USRDTA        647    656
|   D  SYSNAME       657    664
|   D  TIMESTMP      655    672
|   D  OUTQ          673    682
|   D  OUTQLIB       683    692
| D*****
| D OUTDS            DS
| D  DISPOSTN        1      1
| D  CALLAGIN        2      2
| D  RES2            3      4
| D  MSGLEN          5      8B 0
| D  ADDRLEN         9     12B 0
| D  MSGTEXT        13     267
| D  RES3           268    268
| D  EXTOFF         269    272B 0
| D  CCSID          273    276B 0
| D  RES4           277    287

```

```

| D ADDRESS                288   543
| D*
| D EXTLEN                 544   547B 0
| D SUBOFF                548   551B 0
| D SUBLEN                552   555B 0
| D RPLYOFF               556   559B 0
| D RPLYLEN              560   563B 0
| D CCOFF                 564   567B 0
| D CCLEN                 568   571B 0
| D BCCOFF               572   575B 0
| D BCLEN                 576   579B 0
| D BDYPTHOFF            580   583B 0
| D DIRPTHOFF            584   587B 0
| D DIRPTHLEN            588   591B 0
| D ATTPTHOFF            592   595B 0
| D EXTSUBJ              596   605
| D EXTRPLY             606   630
| D EXTBC                631   655
| D EXTCC                656   680
| D BDYLEN              681   684B 0
| D BDYNUMB             685   688B 0
| D BDYX1               689   692B 0
| D BDYY1              693   696B 0
| D BDYOFF1            697   700B 0
| D BDYPLEN1           701   704B 0
| D BDYUSE1            705   705
| D BDYRES1            706   708 0
| D BDYPH1            709   724
| D BDYX2              725   728B 0
| D BDYY2              729   732B 0
| D BDYOFF2            733   736B 0
| D BDYPLEN2           737   740B 0
| D BDYUSE2            741   741
| D BDYRES2            742   744 0
| D BDYPH2            745   760
| D BDYX3              761   764B 0
| D BDYY3              765   768B 0
| D BDYOFF3            769   772B 0
| D BDYPLEN3           773   776B 0
| D BDYUSE3            777   777
| D BDYRES3            778   780 0
| D BDYPH3            781   796
| D DIRPTH            797   811
| D INPUTLEN            DS      4
| D OUTPUTLEN           DS      4
| D OUTINFO            DS      4
| D*****
| C   *ENTRY             PLIST
| C                       PARM                INPUTDS
| C                       PARM                INPUTLEN
| C                       PARM                OUTDS
| C                       PARM                OUTPUTLEN
| C                       PARM                OUTINFO
| C
| C* Check whether there is enough buffer space for all of the data
| C                       EVAL      OUTINFO = X'0000032B'
| C   OUTINFO           IFLE      OUTPUTLEN
| C* If mailtag is TAG001, then use the following data for email
| C   MAILTAG           IFEQ      'TAG001'
| C                       EVAL      ADDRESS = '''nobody@nowhere.com'''
| C* Mail the file
| C                       EVAL      DISPOSTN = '1'
| C* Dont call this pgm again for this file
| C                       EVAL      CALLAGIN = '0'
| C                       EVAL      RES2 = X'0000'
| C* Set up message
| C                       EVAL      MSGLEN = 255

```

```

| C          EVAL      ADDRLEN  = 256
| C          EVAL      MSGTEXT  = 'did you get this?'
| C          EVAL      RES3     = X'00'
| C          EVAL      EXTOFF   = 543
| C          EVAL      CCSID    = 0
| C          EVAL      RES4     = X'000000000000000000000000'
| C          EVAL      EXTLEN   = 52
| C* Set up subject
| C          EVAL      SUBOFF   = 595
| C          EVAL      SUBLEN   = 10
| C          EVAL      EXTSUBJ  = 'TESTING123'
| C* Set up reply to addresses
| C          EVAL      RPLYOFF  = 605
| C          EVAL      RPLYLEN  = 25
| C          EVAL      EXTRPLY  = '''user2@x'''
| C* Set up BCC addresses
| C          EVAL      BCCOFF   = 630
| C          EVAL      BCCLLEN  = 25
| C          EVAL      EXTBCCL  = '''user3@y.c'''
| C* Set up CC addresses
| C          EVAL      CCOFF    = 655
| C          EVAL      CCLEN    = 25
| C          EVAL      EXTCC    = '''user4@y.com'''
| C* Set up a body file
| C          EVAL      BDYPHOFF = 680
| C          EVAL      BDYLEN   = 116
| C          EVAL      BDYNUMB  = 3
| C          EVAL      BDYX1    = 36
| C          EVAL      BDYY1    = 20
| C          EVAL      BDYOFF1  = 20
| C          EVAL      BDYPLEN1 = 10
| C* Set to use a directory with 1st body file
| C* Dont use leading slash
| C          EVAL      BDYUSE1  = '1'
| C          EVAL      BDYRES1  = 0
| C          EVAL      BDYPTH1  = 'Google.htm'
| C* Set up a 2nd body file
| C          EVAL      BDYX2    = 36
| C          EVAL      BDYY2    = 20
| C          EVAL      BDYOFF2  = 20
| C          EVAL      BDYPLEN2 = 14
| C* Dont use directory with 2nd body file
| C          EVAL      BDYUSE2  = '0'
| C          EVAL      BDYRES2  = 0
| C          EVAL      BDYPTH2  = '/More/junk.htm'
| C* Set up a 3rd body file
| C          EVAL      BDYX3    = 36
| C          EVAL      BDYY3    = 20
| C          EVAL      BDYOFF3  = 20
| C          EVAL      BDYPLEN3 = 7
| C* Use directory with 3rd body file
| C* Dont use leading slash
| C          EVAL      BDYUSE3  = '1'
| C          EVAL      BDYRES3  = 0
| C          EVAL      BDYPTH3  = 'pcl.txt'
| C* Set up the directory to use with body files
| C          EVAL      DIRPHOFF = 796
| C          EVAL      DIRPTHLEN = 5
| C          EVAL      DIRPTH   = '/most'
| C* If not TAG001, then don't do any emailing
| C          ELSE
| C          EVAL      DISPOSTN = '0'
| C          ENDIF
| C          ENDIF
| C          SETON

```

LR

---

## Exit Program in C

This is a sample exit program that does not map mail tags, it is an example program intended to illustrate how to use the exit program. You need to customize this exit program to fit your environment before using it. The exit program helps you customize the PDF subsystem. For information about using the PDF subsystem, see Chapter 6, “Using the PDF Subsystem” on page 93. For information about the exit program, see Appendix B, “PDF Mapping Program” on page 143.

```
/*-----*/
/*---
/*--- Program Name: emailmappgm
/*---
/*--- Program Desc: This is a sample user exit for PDF and email function.
/*--- The program requests a larger buffer, and then sets
/*--- up the data area.
/*---
/*--- Invocation: See calling convention below.
/*---
/*---
/*---
/*-----*/

/*-----*/
/*---
/*--- Calling convention:
/*--- Pointer to Email exit input information.
/*--- Length of input information
/*--- Pointer to Email exit output information.
/*--- Length of output information buffer
/*--- Length of output information available
/*---
/*-----*/

/*-----*/
/*--- Includes
/*-----*/

#include stdio.h
#include fstream.h
#include stdlib.h
#include string.h
#include ctype.h
#include errno.h
#include fstream.h
#include epqmapxt.cleinc

#define MAIL_ADDR_DATA_LENGTH 1024

static int firstCall = 1;

/*-----*/
/*--- Structures
/*-----*/

typedef _Packed struct output {
    Qpq_Email_Exit_O_t exitOutputInfo;
    char emailAddrData[MAIL_ADDR_DATA_LENGTH];
} exitOutput;

typedef _Packed struct extension {
    Qpq_Email_Exit_E_t exitextensionInfo;
    char Reply_To[256];
    char CC_Email_Addresses[256];
```

```

    char BCC_Email_Addresses[256];
    char Subject_String[256];
    char Attachments_Directory[256];
} exitExtension;

typedef _Packed struct singleStreamFile {
    Qpq_Stream_File_t streamFileStruct;
    char Path_Name[256];
} singleStreamFile;

typedef _Packed struct allStreamFiles {
    Qpq_All_Stream_Files_t allStmfs;
    singleStreamFile streamFiles[256];
} allStreamFiles;

typedef _Packed struct allOutput {
    exitOutput output;
    exitExtension extension;
    allStreamFiles allStreams;
    allStreamFiles bodyStreams;
} exitAllOutput;

int main(int argc, char* argv[])
{
    Qpq_Email_Exit_I_t *input;
    exitAllOutput* allOutput;

    input = ( Qpq_Email_Exit_I_t* ) argv[1];
    allOutput = (exitAllOutput *)argv[3];
    int* outputInfoBufferSize = (int*) argv[4];
    int* outputInfoSize = (int*) argv[5];

    int streamFileNumber = 0;
    int bodyFileNumber = 0;

/*-----*/
/*--- Set all values to 0 or NULL -----*/
/*-----*/

    allOutput->output.exitOutputInfo.PDF_File_Disposition[0] = '1';
    allOutput->output.exitOutputInfo.Call_Exit_Program_Again[0] = '0';
    allOutput->output.exitOutputInfo.Offset_To_Extension_Area = 0;
    allOutput->output.exitOutputInfo.Message_Text_Length = 0;
    allOutput->output.exitOutputInfo.CCSID_Message_Text_And_Subject = 0 ;
    for (int i = 0; i < 2; i++) allOutput->output.exitOutputInfo.Reserved1[i] = '\0';
    allOutput->output.exitOutputInfo.Reserved2[0] = '\0';
    for ( i = 0; i < 11; i++) allOutput->output.exitOutputInfo.Reserved3[i] = '\0';
    allOutput->extension.exitextensionInfo.Length_ReplyTo = 0;
    allOutput->extension.exitextensionInfo.Length_BCC = 0;
    allOutput->extension.exitextensionInfo.Length_CC = 0 ;
    allOutput->extension.exitextensionInfo.Length_Extension_Area = 52;
    allOutput->extension.exitextensionInfo.Length_Subject = 0;
    allOutput->extension.exitextensionInfo.Offset_Attachments = 0;
    allOutput->extension.exitextensionInfo.Offset_ReplyTo = 0;
    allOutput->extension.exitextensionInfo.Offset_CC = 0;
    allOutput->extension.exitextensionInfo.Offset_BCC = 0;
    allOutput->extension.exitextensionInfo.Offset_Subject = 0;
    allOutput->extension.exitextensionInfo.Offset_Attachments_Directory = 0;
    allOutput->extension.exitextensionInfo.Offset_Path_Body = 0;

/*-----*/
/*--- Check size of buffer that is available for information returned from the -----*/
/*--- PDF mapping program. If the space is too small, return, requesting more. -----*/
/*-----*/
    if (*outputInfoBufferSize > 1000000) {
        *outputInfoSize = 1000000 ;
        firstCall = 0;
    }
}

```



```

    return 0;
}

/* TO Data */

strcpy(allOutput->output.emailAddrData, "noway@jose.com");
allOutput->output.exitOutputInfo.Mail_address_data_length = strlen(allOutput->output.emailAddrData);

/* Reply To Data */

strcpy(allOutput->extension.Reply_To, "noway@jose.com");
allOutput->extension.exitextensionInfo.Length_ReplyTo = strlen(allOutput->extension.Reply_To);
allOutput->extension.exitextensionInfo.Offset_ReplyTo =
    allOutput->output.exitOutputInfo.Offset_To_Extension_Area + sizeof(allOutput->extension.exitextensionInfo);

/* Subject */

strcpy(allOutput->extension.Subject_String, "This is an email" );
allOutput->extension.exitextensionInfo.Length_Subject = strlen(allOutput->extension.Subject_String);
allOutput->extension.exitextensionInfo.Offset_Subject =
    allOutput->output.exitOutputInfo.Offset_To_Extension_Area +
    sizeof(allOutput->extension.exitextensionInfo) + 256 + 256 + 256;

/* CCs */

strcpy(allOutput->extension.CC_Email_Addresses , "CCs@noway.com" ) ;
allOutput->extension.exitextensionInfo.Length_CC = strlen(allOutput->extension.CC_Email_Addresses);
allOutput->extension.exitextensionInfo.Offset_CC =
    allOutput->output.exitOutputInfo.Offset_To_Extension_Area +
    sizeof(allOutput->extension.exitextensionInfo) + 256;

/* BCCs */

strcpy(allOutput->extension.BCC_Email_Addresses, "BCCs@noway.com");
allOutput->extension.exitextensionInfo.Length_BCC = strlen(allOutput->extension.BCC_Email_Addresses);
allOutput->extension.exitextensionInfo.Offset_BCC =
    allOutput->output.exitOutputInfo.Offset_To_Extension_Area +
    sizeof(allOutput->extension.exitextensionInfo) + 256 + 256;

/* Attachment Directory */

strcpy(allOutput->extension.Attachments_Directory , "/home/MyAttachments");
allOutput->extension.exitextensionInfo.Length_Attachments_Directory =
    strlen(allOutput->extension.Attachments_Directory );
allOutput->extension.exitextensionInfo.Offset_Attachments_Directory =
    allOutput->output.exitOutputInfo.Offset_To_Extension_Area +
    sizeof(allOutput->extension.exitextensionInfo) + 256 + 256 + 256 +256;

/* CCSID to use */

allOutput->output.exitOutputInfo.CCSID_Message_Text_And_Subject = 37;

/* Email Disposition */

allOutput->output.exitOutputInfo.PDF_File_Disposition[0] = '1';

/* More Processing Flag */

allOutput->output.exitOutputInfo.Call_Exit_Program_Again[0] = '0';

/* Message Text Data */

strcpy(allOutput->output.exitOutputInfo.Message_text_data, " Please read the attached PDF File " );
allOutput->output.exitOutputInfo.Message_Text_Length = 255;

/* Indicate an IFS file to embed in the Body of the Email */

```

```

strcpy ( allOutput->bodyStreams.streamFiles[bodyFileNumber].Path_Name , "home/MyEmbeddedFiles/MyEmbedFile.txt" );
allOutput->bodyStreams.streamFiles[bodyFileNumber].streamFileStruct.Length_Path_Name =
    strlen(allOutput->bodyStreams.streamFiles[bodyFileNumber].Path_Name ) ;
allOutput->bodyStreams.streamFiles[bodyFileNumber].streamFileStruct.Use_Specified_Directory = '1' ;
allOutput->bodyStreams.allStmfs.Total_Length += allOutput->
    bodyStreams.streamFiles[bodyFileNumber].streamFileStruct.Total_Length;
allOutput->bodyStreams.streamFiles[bodyFileNumber].streamFileStruct.Offset_Path_Name = 20 ;
allOutput->bodyStreams.streamFiles[bodyFileNumber].streamFileStruct.Length_Format_Structure = 20;
allOutput->bodyStreams.allStmfs.Total_Length += allOutput->
    bodyStreams.streamFiles[bodyFileNumber].streamFileStruct.Length_Format_Structure;
allOutput->extension.exitextensionInfo.Offset_Path_Body =
    allOutput->output.exitOutputInfo.Offset_To_Extension_Area + sizeof(allOutput->extension) +
    sizeof (allOutput->allStreams);
for (i = 0; i < 3; i ++) allOutput->
    bodyStreams.streamFiles[bodyFileNumber].streamFileStruct.Reserved3[i] = '\0';
bodyFileNumber++;

/* Indicate an IFS file to attach to the Email */

strcpy ( allOutput->allStreams.streamFiles[streamFileNumber].Path_Name ,
    "home/MyEmbeddedFiles/CEO_Message.wav" );
allOutput->allStreams.streamFiles[streamFileNumber].streamFileStruct.Length_Path_Name =
    strlen(allOutput->allStreams.streamFiles[streamFileNumber].Path_Name ) ;
allOutput->allStreams.streamFiles[streamFileNumber].streamFileStruct.Use_Specified_Directory = '1' ;
allOutput->allStreams.allStmfs.Total_Length += allOutput->
    allStreams.streamFiles[streamFileNumber].streamFileStruct.Total_Length;
allOutput->allStreams.streamFiles[streamFileNumber].streamFileStruct.Offset_Path_Name = 20 ;
allOutput->allStreams.streamFiles[streamFileNumber].streamFileStruct.Length_Format_Structure = 20;
allOutput->allStreams.allStmfs.Total_Length += allOutput->
    allStreams.streamFiles[streamFileNumber].streamFileStruct.Length_Format_Structure;
allOutput->extension.exitextensionInfo.Offset_Attachments =
    sizeof(allOutput->output) + sizeof(allOutput->extension);
for (i = 0; i < 3; i ++) allOutput->
    allStreams.streamFiles[streamFileNumber].streamFileStruct.Reserved3[i] = '\0';
streamFileNumber++;

allOutput->output.exitOutputInfo.Offset_To_Extension_Area = sizeof(allOutput->output) ;
allOutput->bodyStreams.allStmfs.Number_Stream_Files = bodyFileNumber;
allOutput->allStreams.allStmfs.Number_Stream_Files = streamFileNumber;

return 0;
}

```

IBM grants you a nonexclusive copyright license to use all programming code examples from which you can generate similar function tailored to your own specific needs.

All sample code is provided by IBM for illustrative purposes only. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

All programs contained herein are provided to you "AS IS" without any warranties of any kind. The implied warranties of non-infringement, merchantability and fitness for a particular purpose are expressly disclaimed.

## Appendix D. Exit Points

When you install Infoprint Server, it registers two exit points. If these exit points are modified, Infoprint Server might not function properly. This appendix identifies the exit points that Infoprint Server registers and lists their default settings. To view and edit the registry information, use Work With Registry Information (WRKREGINF).

### QIBM\_QPQ\_TRANSFORM

To view and edit the registry information for this exit point, enter WRKREGINF FORMAT(XPDF0100) and then choose option 8 "Work with exit programs". This exit point contains these default values:

Table 19. QIBM\_QPQ\_TRANSFORM Default Values

Exit Program	Exit Program Number	Library	Exit program data length	Exit program data
QIPSINIT	9974	QIPS	11	IPDS PDF

### QIBM\_QIMG\_TRANSFORMS

To view and edit the registry information for this exit point, enter WRKREGINF FORMAT(XFRM0100) and then choose option 8 "Work with exit programs". This exit point contains these default values:

Table 20. QIBM\_QIMG\_TRANSFORMS Default Values

Exit Program	Exit Program Number	Library	Exit program data length	Exit program data
QXTRCLIENT	5380	QIPS	16	PCL <sup>1</sup> AFPDS <sup>2</sup>
QXTRCLIENT	5381	QIPS	16	PS <sup>1</sup> AFPDS <sup>2</sup>
QXTRCLIENT	5382	QIPS	16	PDF <sup>1</sup> AFPDS <sup>2</sup>

#### Notes:

1. Eight bytes long, starting on byte 0.
2. Eight bytes long, starting on byte 7.



---

## Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10594-1785  
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation  
Licensing  
2-31 Roppongi 3-chome, Minato-ku  
Tokyo 106, Japan

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:**

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation  
Mail Drop 001W  
Boulder, CO 80301  
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

IBM grants you a nonexclusive copyright license to use all programming code examples from which you can generate similar function tailored to your own specific needs.

All sample code is provided by IBM for illustrative purposes only. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

All programs contained herein are provided to you "AS IS" without any warranties of any kind. The implied warranties of non-infringement, merchantability and fitness for a particular purpose are expressly disclaimed.

---

## Trademarks

The following are trademarks or registered trademarks of the IBM Corporation:

Advanced Function Presentation	Intelligent Printer Data Stream
AFP	IPDS
AIX	iSeries
AS/400	OfficeVision/400
ESCON	OS/400
IBM	Print Services Facility
Infoprint	Redbooks

The following terms appear in this publication and are trademarks of other companies:

Java is a registered trademark of Sun Microsystems, Inc..

Microsoft®, Windows and Windows NT are registered trademarks of Microsoft Corporation.

Other company, product, and service names may be trademarks or service marks of others.





---

# Glossary

This glossary defines technical terms and abbreviations that are used in Infoprint Server documentation. If you do not find the term you are looking for, refer to the index of this publication or view *IBM Dictionary of Computing*, located at: <http://www.ibm.com/networking/nsg/nsgmain.htm>.

Definitions reprinted from the *American National Dictionary for Information Processing Systems* are identified by the symbol (A) following the definition.

Definitions reprinted from a published section of the International Organization for Standardization (ISO) *Vocabulary—Information Processing* or from a published section of *Vocabulary—Office Machines* developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Committee (ISO/IEC JTC1/SC1) are identified by the symbol (I) following the definition. Because many ISO definitions are also reproduced in the *American National Dictionary for Information Processing Systems*, ISO definitions may also be identified by the symbol (A).

Definitions that are specific to IBM products are so labeled—for example, “In SNA,” or “In the 3820 printer.”

These cross-references are used in this glossary:

- **Contrast with.** Refers to a term that has an opposite or substantively different meaning.
- **See.** Refers to multiple-word terms in which this term appears.
- **See also.** Refers to related terms that have similar, but not synonymous, meanings.
- **Synonymous with.** Appears in the commentary of a preferred term and identifies less desirable or less specific terms that have the same meaning.

## A

**Advanced Function Presentation (AFP).** A set of licensed programs, together with user applications, that use the all-points-addressable concept to print on presentation devices. AFP includes creating, formatting, archiving, retrieving, viewing, distributing, and printing information. See *presentation device*.

**AFP.** See *Advanced Function Presentation*.

**AFP data stream.** A presentation data stream that is processed in the AFP environment. MO:DCA-P is the strategic AFP interchange data stream. IPDS is the strategic AFP printer data stream.

**AFP Viewer.** (1) Refers to the AFP Workbench Viewer and the AFP Viewer Plug-in. A Windows IBM-licensed PC product that lets you see AFP output in a WYSIWYP (what-you-see-is-what-you-print) format. (2) A Windows platform for the integration of AFP-enabling applications and services.

**AFPDS.** A term formerly used to identify the composed page; MO:DCA-P-based data stream that is interchanged in AFP environments.

**anchor point.** The point in a document that signals the beginning of a group of pages to CRTAFPDTA. After the anchor point, CRTAFPDTA adds indexing structured fields to delineate this group.

**architecture.** The set of rules and conventions that govern the creation and control of data types such as text, image, graphics, font, fax, color, audio, bar code, and multimedia.

**ASCII.** American National Standard Code for Information Interchange data encoding, which is the normal (default) type of data encoding in an AIX environment. Contrast with *EBCDIC*.

## B

**Bar Code Object Content Architecture (BCOCA).** An architected collection of control structures used to interchange and present bar code data.

**BCOCA.** See *Bar Code Object Content Architecture*.

**bin.** A paper supply on a cut-sheet printer. See also *cassette*.

**bitmap font.** See *raster font*.

## C

**carriage control character.** An optional character in an input data record that specifies a write, space, or skip operation.

**cassette.** In a cut-sheet printer, a movable enclosure for paper supply. See also *bin*.

**ccsid.** See *coded character set identifier*.

**character.** (1) A symbol used in printing. For example, a letter of the alphabet, a numeral, a punctuation mark, or any other symbol that represents information. (2) A byte of data.

**character set.** (1) A finite set of characters upon which agreement has been reached and that is considered complete for some purpose; for example, each of the characters in ISO Recommendation R646 “6- and 7-bit coded character sets for information processing interchange.” (2) For page printers, the font library member that contains the character graphics and their descriptions.

**coded character set identifier.** The ID associated with a code page.

**coded font.** A font library member that associates a code page and a font character set. For double-byte fonts, a coded font associates more than one pair of code pages and font character sets.

**code page.** A font component that associates code points with character identifiers. A code page also identifies how undefined code points are handled.

**code point.** A one-byte code representing one of 256 potential characters.

**concatenate.** (1) To link together. (2) To join two character strings.

**concatenated data set.** In iSeries, a group of logically connected data sets that are treated as a single data set for the duration of a job step. See also *data set*.

**continuous forms paper.** A series of connected forms that feed continuously through a printing device. The connection between the forms is perforated to enable a user to tear them apart. Before printing, the forms are stacked, folded along the perforations. Contrast with *cut sheet paper*.

**control character.** A character that starts, changes, or stops any operation that affects recording, processing, transmitting, or interpreting data (such as carriage return, font change, and end of transmission).

**cut sheet paper.** Paper that is cut into separate sheets before it is printed on. Contrast with *continuous forms paper*.

## D

**data set.** A named set of records that are stored and processed as a unit. Synonym for *file*.

**data stream.** (1) All information (data and control commands) sent over a data link, usually in a single read or write operation. (2) A continuous stream of data elements being transmitted, or intended for transmission, in character or binary-digit form, using a defined format.

**default.** An attribute, value, or option that is assumed when none is explicitly specified. (l)

**disabled mechanism.** PSF support that enables jobs to print with alternative options if the printer selected for the job does not support a requested option. Contrast with *enabled*.

**document.** (1) A publication or other written material pertaining to a specific subject or related subjects. (2) In word processing, a collection of one or more lines of text that can be named and stored as a separate entity.

**double-byte coded font.** A font in which each character is defined by two bytes. The first byte defines a coded font section and the second defines a code point in that section. Double-byte coded fonts are needed for the support of languages that require more than 256 graphic characters. Two bytes are required to identify each graphic character. Kanji is printed by using a double-byte font. Contrast with *single-byte coded font*.

**download.** To transfer data from a processing unit to an attached device such as a microcomputer for processing.

**duplex printing.** Printing on both sides of a sheet of paper. Contrast with *simplex printing*.

## E

**e-mail address.** The address to which e-mail is sent. A valid e-mail address for Infoprint Server for iSeries is in the form *name@domain* and cannot have extra spaces.

**e-mail tag.** See *mail tag*

**EBCDIC.** Extended binary-coded decimal interchange code.

**electronic overlay.** A collection of constant data, such as lines, shading, text, boxes, or logos, that is electronically composed in the host processor and stored in a library. It can be merged with variable data during printing. Contrast with *page segment*.

**enabled.** (1) Pertaining to a state of the processing unit that allows certain types of interruption. (2) A condition of the printer (physically selected) in which the printer is available to the host processor for normal work. Contrast with *disabled mechanism*.

**end-user interface.** A method by which a customer can obtain the services of a product, for example, coding samples, commands and command lists. Not every product has an end-user interface. Some products provide their services through programming interfaces, some provide services through a command line interface, and others provide services only to other products.

**escape character.** The control character X'2BD3' in a text-control sequence that indicates the beginning of the sequence and the end of any preceding text.

**euro.** The monetary unity of the European Monetary Union (EMU), introduced alongside national currencies on the first of January, 1999.

**EuroReady product.** A product is EuroReady if the product, when used in accordance with its associated documentation, is capable of correctly processing monetary data in the euro denomination, respecting the euro currency formatting conventions (including the euro sign). This assumes that all other products (for example, hardware, software, and firmware) that are used with this product are also EuroReady. IBM hardware products that are EuroReady might or might not have an engraved euro sign key on their keyboards.

**exception.** A condition that exists when the printer:

- Detects an invalid or unsupported command, order, control, or parameter value from the host
- Finds a condition of which the host system must be notified
- Detects a condition that requires the host system to resend data

**exception highlighting.** The markings placed on the printed page to indicate the location of an error in the data stream.

**execution.** The process of carrying out an instruction or instructions of a computer program by a computer. (I) (A)

**extended binary-coded decimal interchange code (EBCDIC).** A coded character set of 256 eight-bit characters.

## F

**font.** A family or assortment of characters of a given size and style; for example, 9-point Bodoni Modern. (A)

**font character set.** Synonym for *character set*.

**form.** A division of the physical medium. Multiple forms can exist on a physical medium. For example, a roll of paper might be divided by a printer into rectangular pieces of paper, each representing a form. An envelope is an example of a physical medium that has only one form. The IPDS architecture defines 4 types of form: cut-sheets, continuous forms, envelopes, and computer output on microfilm. Each type of form has a top edge, a front side, and a back side. Synonymous with *sheet*.

**format.** (1) A specified arrangement of such things as characters, fields, and lines, usually used for displays, printouts, or files. (2) To arrange such things as characters, fields, and lines. (3) To prepare a document for printing in a specified format.

**form definition.** A resource that PSF uses to define the characteristics of a form. It specifies overlays to be used (if any), paper source (for cut-sheet printers), duplex printing, text suppression, the position of MO:DCA data on the form, and the number of copies and modifications of a page.

## G

**GOCA.** See *Graphic Object Content Architecture*.

**Graphic Object Content Architecture (GOCA).** An architecture that provides a collection of graphics values and control structures used to interchange and present graphics data.

**GIF.** See *graphical image format*.

**graphical image format (GIF).** A digital format that is used to compress and transfer graphical information over computer networks. For example, GIF is a common format for graphical information on the Internet.

**group.** A named collection of sequential pages that form a logical subset of a document.

## H

**hardcopy.** (1) A copy of a display image that is generated on an output device such as a printer and can be carried away. (T) (2) A printed copy of machine output in a visually readable form, for example, printed reports, listings, documents, and summaries.

**hexadecimal.** Pertaining to a numbering system with base of 16. Valid numbers use the digits 0 through 9 and characters A through F, where A represents 10 and F represents 15.

**host font.** See *host resource*.

**host processor.** The processing unit to which the page printers are attached through a data-transfer interface.

**host resource.** A resource found in a system library, in a user library, or inline in the print data set.

**host system.** (1) A data processing system that prepares programs and operating environments for another computer or controller. (2) The data processing system to which a network is connected and with which the system can communicate.

## I

**image.** A pattern of toned and untoned pels that form a picture.

**image data.** A pattern of bits, with values of 0 and 1, that defines the pels in an image. (A 1-bit is a toned pel.)

**Image Object Content Architecture (IOCA).** An architected collection of constructs used to interchange and present images.

**indexing.** In CRTAFPDTA, a process of matching reference points within a file and creating structured field tags within the MO:DCA document and the separate index object file.

**indexing with data values.** The process of adding indexing tags to a MO:DCA document using data that is already in the document. This data must be consistently located in the same place in each group of pages.

**indexing with literal values.** The process of adding indexing tags to a MO:DCA document by assigning literal values as indexing tags. This is done when the document is not organized such that common data is located consistently throughout the document.

**index object file.** A file created by CRTAFPDTA that contains Index Element (IEL) structured fields, which identify the location of the tagged groups in the AFP file. The indexing tags are contained in the Tagged Logical Element (TLE) structured fields.

**Infoprint Manager.** A software component of IBM Infoprint Server for iSeries. Infoprint Manager handles the scheduling, archiving, retrieving, and assembly of a PCL, PDF, or PostScript to AFP transform job and its related resource files.

**IOCA.** See *Image Object Content Architecture*.

**initialize.** (1) In programming languages, to give a value to a data object at the beginning of its lifetime. (I) (2) To set counters, switches, addresses, or the contents of storage to zero or other starting values at the beginning of, or at prescribed points in the operation of, a computer routine. (A) (3) To prepare for use. For example, to initialize a diskette.

**inline.** The direction of successive characters in a line of text. Synonymous with *inline direction*.

**inline direction.** The direction of successive characters in a line of text.

**inline resource.** A resource contained in the print data set.

**input/output (I/O).** Pertaining to a device whose parts can perform an input process and an output process at the same time. (I)

**Intelligent Printer Data Stream (IPDS).** (1) The data stream generated by PSF to send to an IPDS page printer. (2) An all-points-addressable data stream that

enables users to position text, images, and graphics at any defined point on a printed page.

**interface.** A shared boundary. An interface can be a hardware component used to link two devices. It can also be a portion of storage or registers accessed by two or more computer programs.

**I/O.** Input/output.

**IPDS.** See *Intelligent Printer Data Stream*.

| **iSeries Information Center.** A Web site that contains  
| technical information about the iSeries, including  
| instructions for completing tasks and informational  
| topics. You can access the version of the Information  
| Center corresponding to the version of the OS/400  
| operating system you have installed. To access the  
| Information Center, go to  
| <http://www.ibm.com/servers/eserver/iseries/infocenter>.

## J

**JPEG.** A standard format for storing compressed true-color images. "JPEG" represents "Joint Photographic Experts Group."

**Joint Photographic Expert Group (JPEG).** The name of the committee that developed the JPEG standard format.

## L

**library.** A file or a set of related files, for example, a page definition library containing one or more page definition files.

**licensed program.** A utility that performs a function for the user and usually interacts with and relies upon system control programming or some other IBM-provided control program. A licensed program contains logic related to the user's data and is usable or adaptable to meet specific requirements.

**line data.** Data prepared for printing on a line printer such as an IBM 3800 Printing Subsystem Model 1. Line data is usually characterized by carriage control characters and table reference characters. Contrast with *MO:DCA*.

**line printer.** A device that prints a line of characters as a unit. (I) (A) Contrast with *page printer*.

**logical page.** A presentation space. One or more object areas or data blocks can be mapped to a logical page. A logical page is rectangular and has specifiable characteristics such as size, shape, orientation, and offset. Orientation and offset are specified relative to a coordinate system for the medium.

**logical page origin.** (1) The point on the logical page from which the positions of images, graphics, page

overlays, and text with 0-degree inline direction are measured. (2) The point on the logical page represented by Xp=0, Yp=0 in the Xp coordinate system.

## M

**macro.** Synonym for *macroinstruction*.

**macroinstruction.** An instruction that causes the execution of a predefined sequence of instructions.

**mail tag.** A keyword or valid e-mail address (in the form *name@domain*) used with the PDF subsystem. If the mail tag is a keyword, it must be mapped to one or more valid e-mail addresses by a user-defined PDF mapping program.

**mapping program.** See *PDF mapping program*.

**Microfilm device.** An output device that presents a hardcopy on microfilm.

**migration.** Activities that relate to the installation of a new version or release of a program to replace an earlier level. Completion of these activities ensures that the applications and resources on your system will function correctly at the new level.

**Mixed Object Document Content Architecture.** A strategic, architected, device-independent data stream used for interchanging documents.

**MO:DCA.** See *Mixed Object Document Content Architecture*.

**MO:DCA data.** Print data that has been composed into pages. Text-formatting programs such as DCF can produce composed text data that consists entirely of structured fields.

**MO:DCA data page.** A page of print data that consists entirely of structured fields.

**MO:DCA print data set.** A print data set that consists entirely of structured fields.

**MO:DCA-P.** Mixed Object Document Content Architecture for Presentation.

## N

**nested resource.** A resource mapped in an overlay.

| **non-square pixels.** An image that has different  
| resolutions in the X (horizontal) and Y (vertical)  
| directions is said to have non-square pixels.

## O

**outline font.** A font technology in which the graphic character shapes are represented by a series of

mathematical expressions that define the outer edges of the strokes. The resulting graphic character shapes can be either solid or hollow. Outline fonts can be scaled (sized) to any size. The IBM outline font character sets have a prefix of CZ. Contrast with *raster font*.

**overlay.** See *electronic overlay*.

## P

**page.** A collection of data that can be printed on a physical sheet of paper.

**page printer.** Any of a class of printers that accepts MO:DCA-P pages. Contrast with *line printer*.

**page segment.** A resource containing MO:DCA data and images, prepared before formatting and included as part of the input for a print job.

**parameter.** (1) A variable that is given a constant value for a specified application and that may denote the application. (l) (A) (2) An item in a menu for which the user specifies a value or for which the system provides a value when the menu is interpreted. (3) Data passed between programs or procedures.

**PDF mapping program.** An exit program written by a user that can be used to accomplish many tasks. It can interpret mail tags, specify the subject of an e-mail, and add text to the beginning of each e-mail. If you are using an SMTP server to send the e-mail, it has additional capabilities. It can also specify carbon copy (cc) and blind carbon copy (bcc), and return-to e-mail addresses, a file to use as the body of the e-mail, and attachments.

**PDF subsystem.** A component of Infoprint Server that converts a spooled file to PDF. It then passes the PDF to PSF for OS/400. PSF for OS/400 then stores the PDF file as a stream file, spools it on an output queue, or e-mails it. You do not need a PSF for OS/400 license to use the PDF subsystem.

**pel.** See *picture element*.

**physical medium.** A physical entity on which information is presented. Examples of physical media are display screens, paper, foils, microfilm, and labels.

**picture element (pel).** An element of a raster pattern about which a toned area on the photoconductor might appear. See also *raster pattern*.

**pitch.** A unit of measurement for the width of a printed character, reflecting the number of times a graphic character can be set in 1 linear inch. For example, 10-pitch has 10 graphic characters per inch. Contrast with *point*.

**point.** A unit of measurement about 1/72 of an inch. It used to measure the height of a font. Contrast with *pitch*.



**point size.** The height of a font in points.

**presentation device.** A device that produces character shapes, graphics pictures, images, or bar code symbols on a physical medium. Examples of physical media are display screens, paper, foils, microfilm, and labels.

**print job.** The data that a user submits to PSF for printing.

**Print Services Facility (PSF).** PSF is a licensed IBM program that manages and controls the input data stream and output data stream that required by supported IBM page printers. PSF manages printer resources such as fonts, images, electronic forms, form definitions, and page definitions, and provides error recovery for print jobs.

When printing line data, PSF supports external formatting by using page definitions and form definitions. This external formatting extends page printer functions such as electronic forms and use of typographic fonts without any change to applications programs.

**printable area.** The area on a sheet of medium on which print can be placed.

**printer.** A presentation device that produces hardcopy output. See *presentation device*.

**processor.** In a computer, a functional unit that interprets and executes instructions. (I) (A)

**PSF.** See *Print Services Facility*.

## R

**raster font.** A font technology in which the graphic characters are defined directly by the raster bit map. Contrast with *outline font*.

**raster pattern.** A series of picture elements (pels) in scan lines to form an image. See also *page segment*.

**record format line data.** A form of line data where each record is preceded by a 10-byte identifier.

**resolution.** In computer graphics, a measure of the sharpness of an image. It is expressed as the number of lines and columns on the display screen or the number of pels per unit of linear measure.

**resource.** (1) A collection of printing instructions that are used by PSF in addition to the print data set to produce printed output. PSF resources include coded fonts, font character sets, code pages, page segments, overlays, form definitions, and page definitions. (2) Any source of aid that is used for performing a task, such as disk storage space, computer processing time, and communication lines.

**resource name.** The name under which a resource object is stored, the first two characters of which indicate the resource type:

<b>X0-XG,XZ</b>	Coded font
<b>T1</b>	Code page
<b>C0-CG,CZ</b>	Font character set
<b>S1</b>	Page segment
<b>F1</b>	Form definition
<b>P1</b>	Page definition
<b>O1</b>	Overlay
<b>H1</b>	Recommended for microfilm

**rotation.** The number of degrees a graphic character is turned relative to the page coordinates.

## S

**segment.** Synonym for *page segment*.

**sheet.** A division of the physical medium on which data is presented. The IPDS architecture defines 4 types of sheet: cut-sheet forms, continuous forms, envelopes, and computer output on microfilm. Each sheet has a front side and a back side. Some types of media consist of multiple sheets; for example, a roll of continuous forms can be divided at the perforations into rectangular sheets. Each sheet usually has carrier or tractor-feed strips, also. Microfilm is another example of a medium comprising multiple sheets, whereas an envelope has only one sheet. Synonymous with *form*.

**simplex printing.** Printing on one side of the paper. Contrast with *duplex printing*.

**single-byte coded font.** A font in which the characters are defined by a one-byte code point. A single-byte coded font contains only one coded font section. Contrast with *double-byte coded font*.

**spooled file.** A file created by an application program that contains the actual information to be printed. It also contains some of the data that controls the format of the printing.

**structured field.** A self-identifying, variable-length, bounded record that can have a content portion that provides control information, data, or both.

**syntax.** The rules and keywords that govern the use of a programming language.

## T

**table reference characters (TRC).** An optional control character in an input record that identifies the font with which the record is to be printed. The table reference character corresponds to a font number defined in a page definition font list or to the order of font names listed in the CHARS parameter in the JCL.

**tag.** A type of structured field that is used to index an AFP document. Tags associate an index attribute - value pair with a specific page or group of pages in a document.

**Tag Image File Format (TIFF).** A graphic file format used to store and exchange scanned images; compatible with a number of personal computing platforms.

**terminate.** (1) To stop the operation of a system or device. (2) To stop execution of a program.

**text.** A graphic representation of information on an output medium. Text can consist of alphanumeric characters and symbols that are arranged in paragraphs, tables, columns, or other shapes.

**TIFF.** See *Tag Image File Format*.

**token ring.** A network configuration in which tokens are passed in a circuit from node to node. A node that is ready to send can capture the token and insert data for transmission.

**trace.** A record of the execution of a computer program. It exhibits the sequences in which the instructions were executed. (A)

**TRC.** See *table reference character*.

**trigger.** Data values for which CRTAFPDTA searches, to delineate the beginning of a new group of pages. The first trigger is then the anchor point from which CRTAFPDTA locates the defined index values. See *anchor point*.

## U

**upload.** (1) To transfer programs or data from a connected device, typically a personal computer, to a computer with greater resources. (T) (2) To transfer data from a device, such as a workstation or a microcomputer, to a computer. Contrast with *download*.

## V

**Viewer.** See *AFP Workbench Viewer*.

## X

**X-axis.** In printing, an axis perpendicular to the direction in which the paper moves through the printer. See also *Y-axis*.

**X-extent.** A measurement along the X-axis.

## Y

**Y-axis.** In printing, an axis parallel to the direction in which the paper moves through the printer. See also *X-axis*.

**Y-extent.** A measurement along the Y-axis.





---

## Bibliography

This bibliography lists the titles of publications containing additional information about Infoprint Server for iSeries, the OS/400 operating system, Advanced Function Presentation, and related products.

The titles and order numbers may change from time to time. To verify the current title or order number, consult your IBM marketing representative.

You can obtain many of the publications listed in this bibliography from the Printing Systems Digital Library: <http://www.ibm.com/printers/r5psc.nsf/web/manuals> or the Online Publications Website: <http://publib.boulder.ibm.com/>

---

### Infoprint Server

Publication	Order Number
<i>Infoprint Server for iSeries: User's Guide</i>	G544-5775-01
<i>Infoprint Server for iSeries: Introduction and Planning Guide</i>	G544-5774-01

---

### Advanced Function Presentation (AFP)

Publication	Order Number
<i>Guide to Advanced Function Presentation</i>	G544-3876
<i>Printing and Publishing Cluster Collection CD-ROM</i>	SK2T-2921

---

### iSeries Access

Publication	Order Number
<i>iSeries Access for Windows--Setup</i>	SC41-5507-03
<i>LPS: AS/400 Client Access Family for Windows</i>	GC41-5041-02
<i>LPS: Client Access Ultimeia Tools for AS/400</i>	GC41-4075-00

---

### Infoprint Designer

Publication	Order Number
<i>Infoprint Designer for iSeries: Getting Started</i>	G544-5773-00

---

## OS/400

Publication	Order Number
<i>AS/400 Guide to Advanced Function Presentation and Print Services Facility</i>	S544-5319-03
<i>AS/400 Command Language Reference</i>	SC41-3722
<i>DDS Reference</i>	SC41-5712
<i>Printer Device Programming</i>	SC41-5713-05
<i>Software Installation</i>	SC41-5120-06
<i>System API Programming</i>	SC41-5800-00

---

## Printers

Publication	Order Number
<i>IBM Printing Systems: Printer Summary</i>	S544-5749
<i>IBM PagePrinter 3812 Introduction and Planning Guide</i>	G544-3265
<i>IBM LaserPrinter 4028 Introduction and Planning Guide</i>	S544-4258
<i>IOCP and ESCON<sup>®</sup> Reference</i>	GC38-0401

---

## PrintSuite

Publication	Order Number
<i>APU Guide for PrintSuite</i>	S544-5351
<i>SAP/R3 Guide for PrintSuite</i>	S544-5412
<i>Page Printer Formatting Aid: User's Guide</i>	S544-5284-06
<i>AFP Toolbox for Multiple Operating Systems User's Guide</i>	S544-5292

---

## Redbooks

Publication	Order Number
<i>IBM AS/400 Printing V</i>	SG24-2160
<i>IBM @server iSeries Printing VI: Delivering the Output of e-business</i>	SG24-6250

---

## TCP/IP

Publication	Order Number
<i>Internetworking with TCP/IP, Principles, Protocols, and Architecture</i>	SC31-6144
<i>TCP/IP Tutorial and Technical Reference</i>	GG24-3376
<i>TCP/IP Configuration and Reference</i>	SC41-5420-04





---

# Index

## Special characters

-a 41, 59, 77  
-alg 42, 60, 78  
-C 43, 61, 79  
-calib 42, 60, 79  
-choice 79  
-clean 43, 79  
-cmp 43, 61, 80  
-crop 44, 61, 80  
-fit 44, 62, 80  
-force 62, 81  
-gcorr 44, 62, 81  
-ink 45, 81  
-inv 45, 81  
-j 45, 62, 81  
-l 46, 63, 82  
-M 46, 63, 82  
-mp 83  
-ms 46, 63, 83  
-msf 46, 63, 83  
-nomp 83  
-nosniff 47, 83  
-noterm 47, 64, 83  
-nov 47, 64, 84  
-o 47, 64, 84  
-outbits 47, 64, 84  
-outcolor 48, 65, 84  
-p 48, 65, 84  
-pagetype 48, 65, 85  
-paper 48, 65, 85  
-r 49, 66, 85  
-respath 49, 66, 85  
-rot 49, 66, 86  
-scale 49, 66, 86  
-sgcorr 50, 67, 87  
-sniff 47, 83  
-term 47, 64, 83  
-thresh 50, 67, 87  
-v 47, 64, 84  
-w 50, 67, 87  
-wrkdir 68, 87  
-x 51, 68, 88  
-y 51, 68, 88

## Numerics

5.2 new function 5

## A

accessing the PDF subsystem from iSeries Access 98  
ACIF 9  
AFP Toolbox 100  
anchor 20  
archiving  
    retrieving resources for 14

## C

command  
    CRTAFPDTA 7  
    gif2afp 37  
    jpeg2afp 55  
    tiff2afp 73  
configuration file  
    gif2afp 39  
    jpeg2afp 57  
    tiff2afp 75  
configuring an IPDS to PDF printer 96  
Create AFP Data  
    *See also* command  
    authorities needed 11  
    e-mailing the output 32  
    examples 25  
    getting ready to use 11  
    group name 12  
    groups 12  
    indexing with 12  
    indexing with CRTAFPDTA 26  
    merging resources with data 26  
    overview 7  
    parameters 15  
        AFP characters (AFPCHARS) 15  
        Form definition (FORMDF) 15  
        From spooled file (FROMSPLF) 16  
        Generate group names (IDXGRPNAM) 17  
        Generate page names (IDXPAGNAM) 18  
        Image output (IMAGEOUT) 21  
        Index code page identifier (IDXCDEPAG) 16  
        Index field definition (IDXTAGFLD) 19  
        Index tag definition (IDXTAG) 18  
        Index trigger definition (IDXTRG) 20  
        Indexing page limit (IDXPAGLMT) 17  
        Information for index file (IDXOBJ) 17  
        Job name (JOB) 21  
        Page definition (PAGDFN) 22  
        Resource Data (RSCDTA) 22  
        Select index for group names (IDXGRP) 16  
        Spooled file created (CRTDATE) 15  
        Spooled file number (SPLNBR) 23  
        System name (JOBSYSNAME) 21  
        To index stream file (TOIDXSTMF) 23  
        To merged stream file (TOMRGSTMF) 24  
        To resource stream file (TORSCSTMF) 25  
        To stream file (TOSTMF) 25  
    retrieving resources with 14  
    searches for resources 14  
    syntax 9  
    viewing the output 31  
CRTAFPDTA 100

## D

data queue 112

- data transformation
  - halftoning 39, 57, 75
  - scaling 38, 56, 74
- DDS 99

## E

- e-mail
  - enabling your OS/400 for 130
  - sending 129
  - steps to send a PDF file 130
  - tasks 129
  - time stamp 137
- e-mail error 110
- e-mail error messages 136
- editing your Windows path 41, 59, 77
- enabling your OS/400 to send e-mail 130
- environment variables
  - gif2afp 40
  - jpeg2afp 58
  - tif2afp 76
- error messages, e-mail 136
- errors, e-mail 110
- examples
  - e-mailing CRTAFPDТА output 33
  - exit program in C 159
  - GIF to AFP transform 52
  - indexing with CRTAFPDТА 26
  - JPEG to AFP transform 69
  - merging resources with data 26
  - PDF mapping program in C 155
  - PDF mapping program in RPG 156
  - TIFF to AFP transform 89
  - using PDF subsystem to send e-mail 97
- exit points 163
  - QIBM\_QIMG\_TRANSFORMS 163
  - QIBM\_QPQ\_TRANSFORM 163
- exit program template in C 159

## F

- Font Downloader 124
- font mapping file 120
- font sieving 107

## G

- GIF to AFP transform 37
- gif2afp 37
  - authorities 41
  - configuration file 39
  - editing your Windows path 41, 77
  - environment variables 40
  - error recovery 53
  - halftoning 39
  - input to 51
  - installing 39
  - invoking 52
  - limitations 54
  - options and values 41
  - output from 38

- gif2afp (*continued*)
  - overview 37
  - parameters 41
  - planning for 39
  - scaling 38
  - syntax 37

## H

- halftoning
  - gif2afp 39
  - jpeg2afp 57
  - tiff2afp 75

## I

- image configuration object 122
- index object file
  - defined 12
- indexing
  - description 12
  - for viewing 12
  - indexing data for 12
  - limitations 13
- indexing anchor record 20
- indexing with data values 13
- indexing with literal values 13
- Infoprint Designer for iSeries 140
- Infoprint Server
  - components 1
  - installing 6
  - introducing 1
  - transforming PCL, PDF, and PostScript data with 115
- Infoprint Server components
  - CRTAFPDТА 7
  - GIF to AFP transform 37
  - JPEG to AFP transform 55
  - overview of 1
  - PCL, PDF, and PostScript to AFP transform 115
  - TIFF to AFP transform 73
- input
  - CRTAFPDТА 9
  - gif2afp 51
  - jpeg2afp 68
  - tiff2afp 88
- inserting a mail tag in the printer file 133
- inserting a mail tag in the spooled file 134
- installing gif2afp 39
- installing jpeg2afp 57
- installing tiff2afp 75
- iSeries Access 139
  - accessing the PDF subsystem 98
  - overview 139
- iSeries Access for Web 139
- iSeries Information Center xi

## J

- job logs 111
- JPEG to AFP transform 55

- jpeg2afp 55
  - authorities 59
  - configuration file 57
  - editing your Windows path 59
  - environment variables 58
  - error recovery 70
  - halftoning 57
  - input to 68
  - installing 57
  - limitations 71
  - options and values 59
  - output from 56, 68
  - overview 55
  - parameters 59
  - planning for 57
  - scaling 56
  - syntax 55
- jpg2afp
  - invoking 69

## L

- limitations
  - indexing 13

## M

- mail tag 133
  - inserting in the printer file 133
  - inserting in the spooled file 134
- messages, non-delivery 136
- modifying the input file 99

## N

- new in 5.2 5
- non-delivery messages 136

## O

- options
  - gif2afp 41
  - jpeg2afp 59
  - tiff2afp 77
- OS/400 command syntax x
- OS/400 printing 1
- overview of infoprint server components 2
- overview of Infoprint Server components 1

## P

- parameters
  - CRTAFPDTA 15
  - gif2afp 37
  - jpeg2afp 55
  - tiff2afp 73
- PCL, PDF, and PostScript to AFP transform 115
  - authorities needed 117
  - configuring Transform Manager 117
  - creating the work directory 117

- PCL, PDF, and PostScript to AFP transform *(continued)*
  - customizing PDF to AFP transform 120
  - customizing Postscript to AFP transform 120
  - deactivating 123
  - ending Transform Manager 121
  - image configuration object 122
  - increasing the memory for input 121
  - overview 115
  - planning for 116
  - printer device description, setting up 122
  - reactivating 123
  - setting up the printer device description 122
  - starting Transform Manager 121
  - syntax 116
  - system requirements 117
  - transforming data with 122
  - troubleshooting 126
  - using 122
  - using Font Downloader 124
- PDF conversion completion notification 112
- PDF mapping program 143
- PDF mapping program template in C 155
- PDF mapping program template in RPG 156
- PDF subsystem
  - configuring the device 96
  - error messages 110
  - error recovery 110
  - exit program template in C 159
  - example 97
  - file permissions 101
  - font considerations 107
  - modifying the input file 99
  - orientation 112
  - overview 93
  - page size 112
  - PDF mapping program 143
  - PDF mapping program template in C 155
  - PDF mapping program template in RPG 156
  - planning for 95
  - PSF configuration object 95
  - security violation 112
  - start the writer 97
  - transforming data with 97
  - troubleshooting 111
  - usage notes 106
    - bar codes 107
    - color 107
    - error checking 107
    - find function 107
    - fonts 107
    - maximum number of pages 109
    - output file size 108
    - page numbering 109
    - printer features 109
    - printing considerations 109
    - unsupported IPDS functions 108
    - viewing considerations 110
  - vary the device ON 96
- PDF to AFP transform
  - customizing 120

- PostScript to AFP transform
  - customizing 120
- PSF configuration object, setting up 95
- publications, related 177

## Q

- QIBM\_QIMG\_TRANSFORMS 163
- QIBM\_QPQ\_TRANSFORM 163

## R

- related products 139
  - Infoprint Designer for iSeries 140
  - iSeries Access 139
  - iSeries Access for Web 139
- related publications 177
- related tasks 141

## S

- scaling
  - gif2afp 38
  - jpeg2afp 56
  - tiff2afp 74
- sending e-mail 129
- setting QUTCFFSET 137
- starting the writer 97
- stdout 51, 69, 88
- syntax
  - CRTAFPDTA 9
    - explained x
  - gif2afp 37
  - jpeg2afp 55
  - PCL to AFP transform 116
  - PDF to AFP transform 116
  - PostScript to AFP transform 116
  - tiff2afp 73

## T

- tasks
  - checking for non-delivery messages 136
  - creating a PSF configuration object 133
  - e-mail CRTAFPDTA output 32
  - enabling your OS/400 to send e-mail 130
  - generating index tags 99
  - generating multiple PDF files 99
  - indexing with CRTAFPDTA 12
  - Planning for a PCL, PDF, or PostScript Transform 116
  - planning for CRTAFPDTA 11
  - planning for gif2afp 39
  - planning for jpeg2afp 57
  - planning for the PDF subsystem 95
  - planning for tiff2afp 75
  - sending e-mail 129
  - setting QUTCFFSET 137
  - Setting up a POP3 client for a SNDDST sender 136
  - setting up the OS/400 as a POP e-mail client 132

- tasks (*continued*)
  - transforming GIF data to AFP 51
  - transforming JPEG data to AFP 68
  - transforming TIFF data to AFP 88
  - turning a PC AFP resource into an OS/400 resource 141
  - Using AFP Manager to send e-mail 135
    - using CRTAFPDTA 7
    - using gif2afp 37
    - using jpeg2afp 55
    - using mail tags 133
    - using the PDF subsystem 93
    - using tiff2afp 73
- TIFF to AFP transform 73
- tiff2afp 73
  - authorities 77
  - configuration file 75
  - environment variables 76
  - error recovery 90
  - half-toning 75
  - input to 88
  - installing 75
  - invoking 89
  - limitations 90
  - options and values 77
  - output from 74
  - overview 73
  - parameters 77
  - planning for 75
  - scaling 74
  - syntax 73
- time stamp 137
- Transform Manager, configuring 117
- Transform Manager, ending 121
- Transform Manager, starting 121
- transforming
  - GIF data to AFP 37
  - GIF data to PostScript 37
  - JPEG data to AFP 55
  - JPEG data to PostScript 55
  - line data to AFP 7
  - PCL to AFP 115
  - PDF to AFP 115
  - PostScript to AFP 115
  - spooled file to PDF 93
  - TIFF data to AFP 73
  - TIFF data to PostScript 73
- troubleshooting
  - PCL to AFP transform 126
  - PDF subsystem 111
  - PDF to AFP transform 126
  - PostScript to AFP transform 126
- turning a PC AFP resource into an OS/400 resource 141

## U

- using AFP Toolbox 100
- using CRTAFPDTA to generate multiple PDF files 100
- using DDS 99
- using mail tags 133



## **V**

- varying the device ON 96
- viewing
  - retrieving resources for 14

## **W**

- Windows command syntax x
- work directory, creating 117



---

# Readers' Comments — We'd Like to Hear from You

Infoprint Server for iSeries  
User's Guide  
Version 5, Release 2.0

Publication No. G544-5775-02

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Complete	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to find	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to understand	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Well organized	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Applicable to your tasks	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? ☐ Yes ☐ No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

---

Name

---

Address

---

Company or Organization

---

Phone No.



Cut or Fold  
Along Line

Fold and Tape

Please do not staple

Fold and Tape



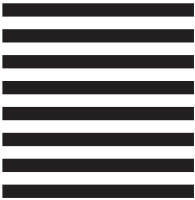
NO POSTAGE  
NECESSARY  
IF MAILED IN THE  
UNITED STATES

**BUSINESS REPLY MAIL**

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation  
IBM Printing Systems Division  
Department H7FE, Building 004M  
Information Development  
P.O. Box 1900  
Boulder, CO USA 80301-9817



Fold and Tape

Please do not staple

Fold and Tape

Cut or Fold  
Along Line





Program Number: 5722-IP1

Printed in U.S.A.

G544-5775-02



Spine information:



Infoprint Server for iSeries

Infoprint Server for iSeries: User's Guide

Version 5,  
Release 2.0